2 p (mix)

NASA CR-122396

Final Report

Metric Analysis of Minitrack Optical and Interferometer Data

Prepared for:

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771

In performance of:

Contract NAS5-10783

Prepared by:

DBA SYSTEMS, INC.
9301 Annapolis Road, Suite 200
Lanham, Maryland 20801

30 June 1971

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Metric Analysis of Minitrack Optical and Interferometer Data		5. Report Date 30 June 1971
		6. Performing Organization Code
7. Author(s) Duane C. Brown, Georg E. Morduch, James B. Willman		8. Performing Organization Report No.
9. Performing Organization Name and Address		10. Work Unit No.
		11. Contract or Grant No. NAS5-10783
12. Sponsoring Agency Name and Address		13. Type of Report and Period Covered Final (1 July 69 - 30 June 71)
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

The Network Analysis Program (NAP-II), which has the capability of simultaneously solving for orbits and tracking station error model terms, was, after several modifications, used in the calibration of Minitrack stations using Minitrack satellite measurements (self-calibration). Several support programs were written to aid in this task.

A simultaneous four-arc solution was obtained. A comparison with optically determined arcs for the same time spans showed RMS position differences of 67m, 86m, 124m and 168m for the 4 arcs considered.

An apparatus incorporating a diffraction grating was designed and successfully used to measure the drift rate of the Fort Myers MOTS camera drive.

17. Key Words (Selected by Author(s))		18. Distribution Statement	
19. Security Classif. (of this report) UNCLASSIFIED	20. Security Classif. (of this page) UNCLASSIFIED	21. No. of Pages	22. Price* $17.75

PREFACE

The objective of this work was to perform a study, prepare a Minitrack error model with as many of the coefficients as practical being established by pre-flight calibration measurements, prepare a computational method to utilize the error model, and to monitor the MOTS camera drive stability using a diffraction grating.

The NAP-II program, which has the capability of simultaneously solving for orbits and tracking station error model terms, was modified to substantially increase its computational speed and substantially reduce its disk memory requirements, thus making it possible to meet the contract objective within the required time-scale.

Twenty-five short optically determined reference arcs were used to calibrate the Fort Myers Minitrack station.

The NAP program was used to obtain a simultaneous four-arc solution (total length 25 days) of the 4 orbits and Minitrack station calibration numbers (for all stations).

A diffraction grating was designed and used for monitoring the stability of the Fort Myers MOTS camera drive.

The multi-arc approach to the self-calibration of Minitrack stations appears to give very good results when judged on the comparison between Minitrack and optically determined orbits. The four arcs processed under this contract showed RMS position differences of 67m, 86m, 124m and 168m, respectively. This compares favorably with an RMS position difference of 165m for the first arc based on "aircraft calibrated" Minitrack measurements.

Use of the diffraction grating apparatus developed under this contract provides a simple and inexpensive means of correcting plate errors caused by the (periodic) instability of the MOTS camera drive.

It is recommended that the approach to Minitrack self-calibration developed under this contract be further tested using other satellites (than GEOS-I) before "aircraft calibration" is finally abandoned.

PRECEDING PAGE BLANK NOT FILMED

CONTENTS

Metric Analysis of Minitrack Optical and Interferometer Data

Appendices

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF ILLUSTRATIONS

Section 1

SECTION 1

MONITORING OF MOTS STABILITY BY MEANS OF A DIFFRACTION GRATING APPARATUS

1.1 INTRODUCTION

The NASA MOTS camera and the U. S. Air Force PC-1000 camera employ the same 1000mm f/5 Baker telephoto lens. The PC-1000 uses an alta-azimuth mount and is locked in a fixed orientation (relative to the earth) throughout each exposure. A precisely timed shutter provides intermittent exposures of star trails to serve as control points. The MOTS camera, by contrast, uses an equatorial mount and is sidereally driven to maintain a fixed orientation with respect to the right ascension declination frame. By virtue of this mode of operation, MOTS does not require a precisely timed shutter and provides a greater abundance of stellar images than a PC-1000. Both MOTS and PC-1000 can potentially produce accuracies of about 0.6 seconds of arc for satellite directions. This potential, however, may not be routinely realized.

One of the key advantages claimed for the PC-1000 (and for the fixed camera mode of operation, in general) is that any significant drift in the orientation of the camera throughout an exposure can be detected (and generally corrected for) by virtue of separate reductions performed on each sequence of stellar exposures. Brown, (Reference 1) reports results of an investigation of the stability of a PC-1000 over a period of about one half hour. Reductions were performed on exposures made at 5 minute intervals, leading to results plotted in Figure 1, which depicts the temporal variation of the three angular elements of orientation about their respective means. The plotted results for hour angle are normalized by the customary process of multiplication by cosine of declination. Each point is accompanied by a vertical bar defining its plus and minus 1 sigma confidence intervals. The 1 sigma values for declination and normalized hour angle are slightly less than 0.2 second of arc; for swing angle, they are

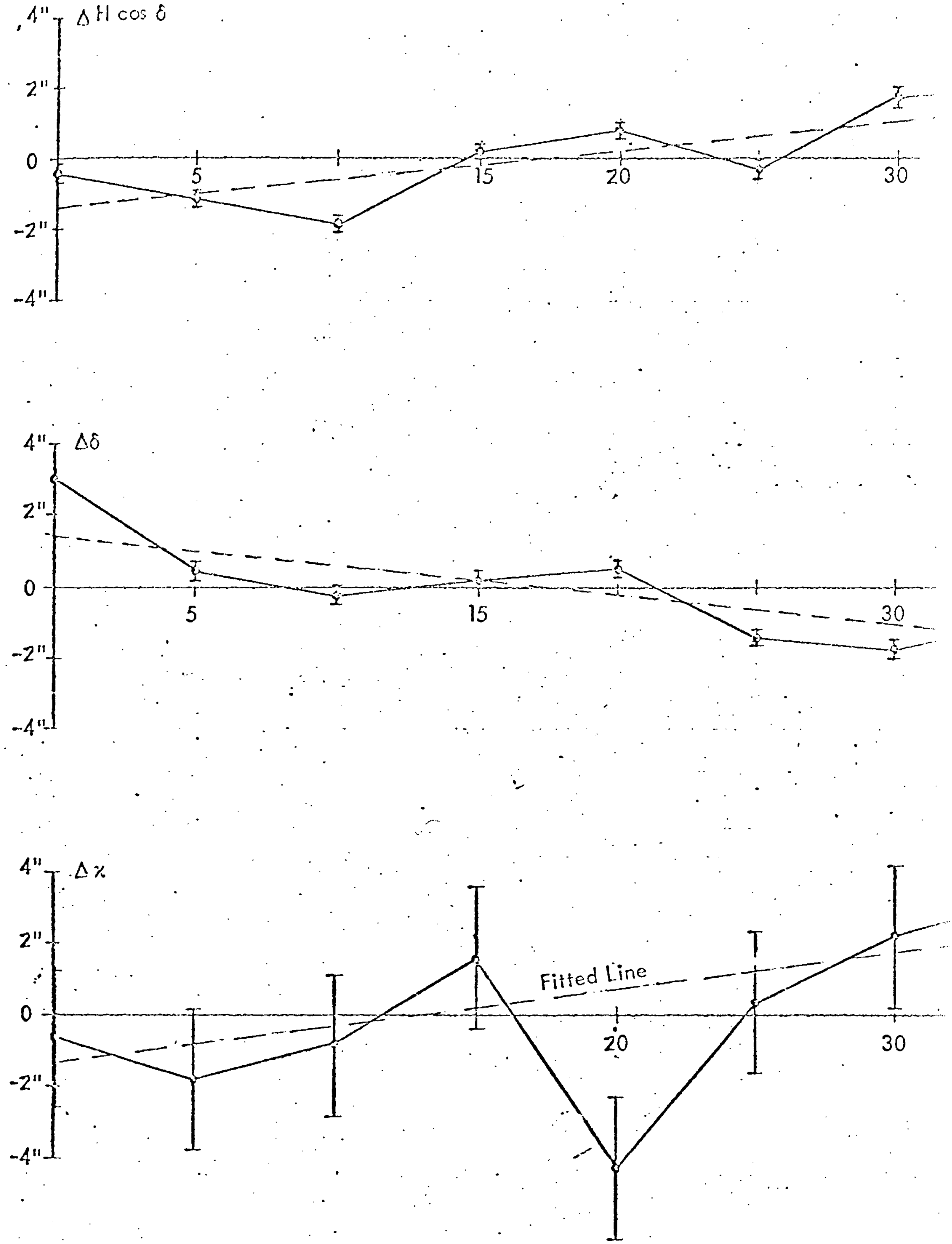


FIGURE 1. Variation in angular stability of PC-1000 camera over period of 35 minute as reported in reference 1.

generally about ten times greater, averaging close to 2 seconds of arc. This disparity in sigmas is attributable to the fact that the focal length (1000mm) of the PC-1000 is about 10 times greater than the semi-diagonal of the plate format. The projective effect of an error of 0.2 second of arc is the direction of the camera axis is equivalent to that of an error of about 1 micron on the plate. By the same token, the projective effect of an error of 2 seconds in swing angle is equivalent to that of an error of about 1 micron near the edge of the plate. Thus, there is no actual projective disparity in the relative sigmas of the angular elements.

Figure 1 clearly demonstrates that significant changes in the orientation of the PC-1000 can occur throughout a period as short as 5 minutes. Indeed, changes in hour angle and declination of as much as 2 seconds of arc in 5 minutes are not unusual. Be this as it may, the fact is that in a vixed camera operation, any changes of significance can be detected and their effects on directions of satellites can be largely removed by an interpolative process. Thus while stability is a problem in PC-1000 operations, it is a problem that can be routinely overcome.

In general, MOTS is subject to the same sources of instability as the PC-1000 (e.g., thermal imbalances, wind loading, etc.). However, MOTS is also subject to drifts induced by the sidereal drive. Whatever their origin, small instabilities in MOTS orientation cannot be routinely detected because their primary effect is merely to cause a slight enlargement in the sizes of stellar images. This is a matter of no consequence insofar as the plate reduction itself is concerned, for instability affects all stars alike and hence does not degrade the residuals produced by the least squares adjustment. However, flashes are affected by the instantaneous rather than the integrated effects of instability. Hence, satellite directions derived from MOTS observations inherit fully the instantaneous departures in orientation of the camera from the mean orientation deduced from the stellar exposures. Clearly then, a simple, practical, routine means for monitoring the stability of MOTS orientation throughout an exposure would constitute a significant advance. The objective of the present investigation is to evaluate an approach that holds promise in this regard.

1.2 GENERAL CHARACTERISTICS OF ERROR INDUCED BY MOTS SIDEREAL DRIVE

A study reported by Harris, Cartwright and Oosterhout (Reference 2) provides a good understanding of the nature of the errors induced by the MOTS sidereal drive. The data analyzed in Reference 2 consists of MOTS images of a stationary collimator generating an artificial star produced by a pinhole illuminated by a strobe light of 1 millisecond duration flashing at 10 second intervals. The exposures of the flashing collimator were made over a period of 16 minutes with the MOTS axis in a nominally horizontal orientation and with the camera being driven in the normal sidereal mode. This generated a set of almost 100 successive images which, with a flawless drive, would ideally be at a constant declination and spaced at equal increments of hour angle. Departures of actual images from their ideal positions provide the desired measure of errors of the drive. In the analysis provided in Reference 1, polynomials in time were fitted to the measured x and y coordinates of the collimator images in order to account for low order effects such as the slight curvature of the trials. The residuals about the fitted polynomials reflect the combined effect of plate measuring errors and higher order sidereal drifts (slow thermal drifts and the like would be absorbed by the fitted polynomials).

Graphs of residuals in right ascension and declination for a particular trial presented in Reference 2 are reproduced in Figures 2 and 3 below. The most significant finging of Reference 1, namely the existence of a well defined periodic drift in right ascension, is well illustrated in Figure 2. The period of the drift is almost precisely 90 seconds and is attributed to the interaction of the worm and sector gears of the drive system. The amplitude of the drift is reported in most cases to range between 2 and 3 arc seconds in right ascension and to be only about half as great in declination.

Because of geometrical considerations, the procedure employed in Reference 1 to monitor MOTS stability is not one that can be practically adopted to effect a routine calibration of drift affecting operational plates. The procedure has served well, however, to define the essential characteristics of MOTS drift, thereby making clear the problem to be overcome by any method designed for routine operational calibration of drift.

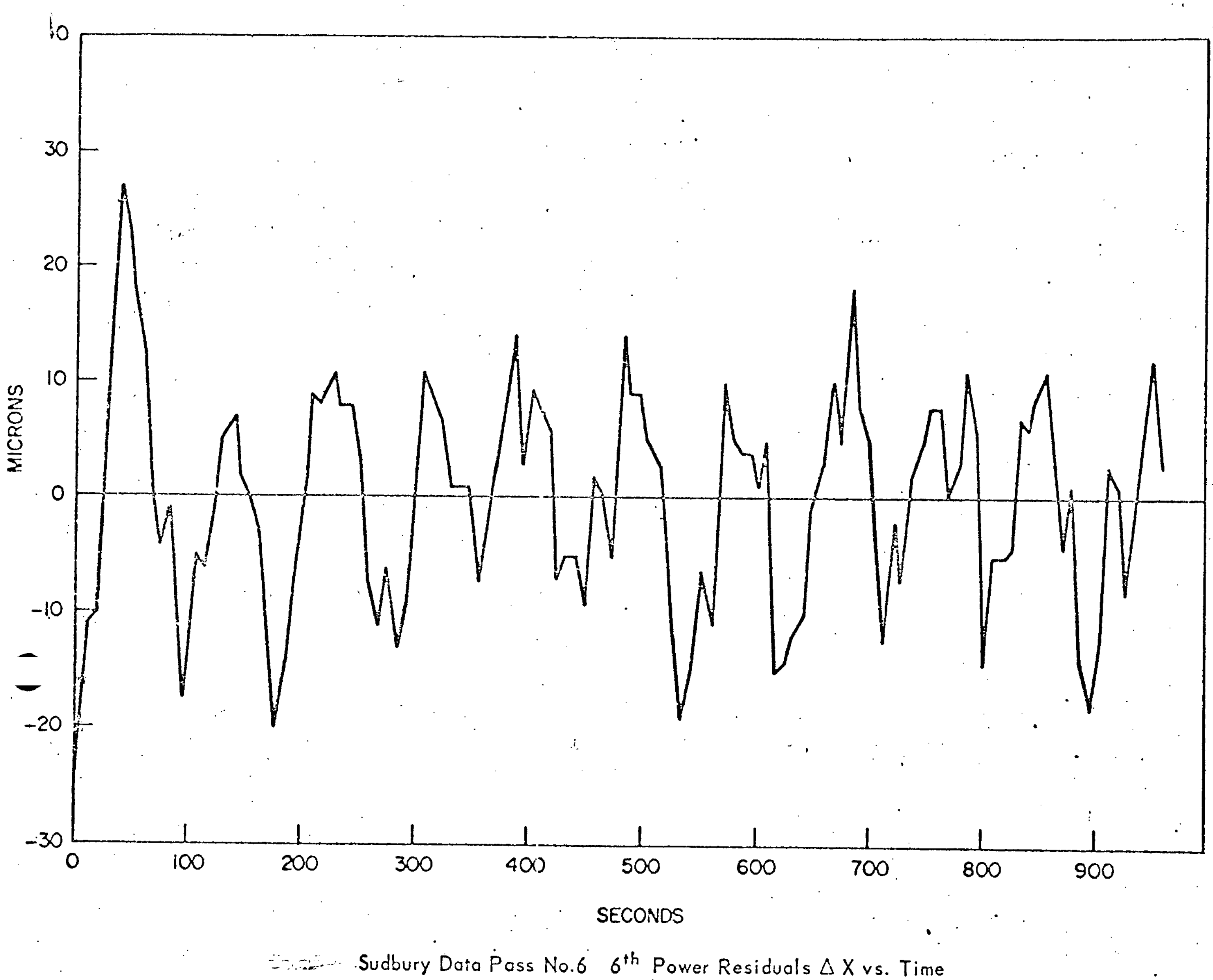


FIGURE 2. Angular stability of MOTS in right ascension as determined by flashing collimator method reported in reference [2] (5 microns = 1 second of arc).

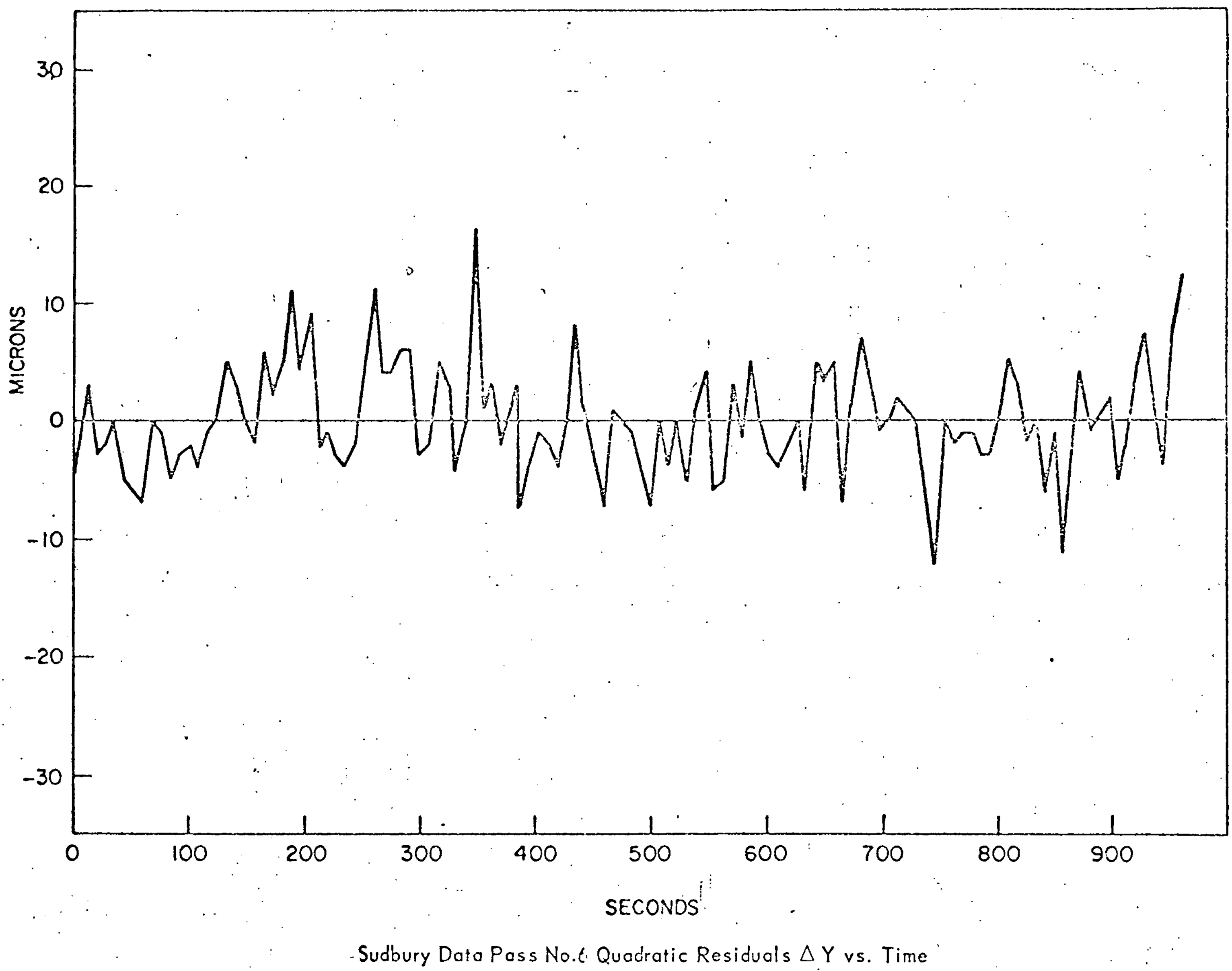


FIGURE 3. Angular stability of MOTS in declination as determined by flashing collimator method reported in reference [2] (5 microns = 1 second of arc).

1.3 THE DIFFRACTION GRATING METHOD

The present investigation is concerned with the evaluation of a method of monitoring MOTS stability proposed to NASA by DBA Systems. As originally conceived, it involved placement of a coarse diffraction grating in front of the MOTS lens to generate measurable first order diffraction images of sufficiently bright stars. A series of six exposures would be made at suitable intervals with orientation of the grating being changed in nominal 15 degree steps between exposures. This sequence of exposures was to provide material for a 'precalibration' of MOTS drift. Shortly before the exposure of the satellite, the diffraction grating would be automatically retracted; immediately following the completion of the satellite observations, the grating would be returned in front of the objective for a series of six 'postcalibration' exposures at 15 degree steps. A ten times enlargement of the image generated by this process for a sufficiently bright star would have the general appearance shown here:  . Now, in theory, the mean of a pair of opposing diffraction images will coincide precisely with the center of the associated primary central image. It follows that, should the central image wander slightly in position (because of drift of the mount), the locus of the wander can be reconstructed from the means of diffraction images taken at different times. The principle of the diffraction grating approach is thus straightforward and simple.

From early experiments, a more satisfactory operational method of implementing the diffraction grating approach evolved. It was learned that it was not really necessary to employ the entire aperture of the MOTS in order to obtain satisfactory images; rather, a diffraction grating having a diameter of less than half that of the MOTS objective could provide altogether acceptable results. This finding meant that the grating could remain in place and undergo its series of step rotations throughout the period in which satellite observations were being made. This obviates the need for interpolation between pre and post-calibration series of exposures and provides instead a running calibration spanning the interval of actual interest.

The final version of the apparatus that evolved from exploratory exercises is shown at half scale in Figure 4. The grating itself is of 3.5-inch diameter and is supported over the center of the field of the objective by a three-legged spider. The grating and spider obscure about 20 percent of the area of the aperture. With the particular grating employed, 25 percent of the light incident on the grating goes into the central image. Accordingly, the effective loss of light due to the imposition of the grating is only about 15 percent. The grating is rotated by means of a belt-driven turntable connected to a motor. A remote manually operated, pushbutton switch causes a spring loaded solenoid plunger to retract from one of a series of slots spaced at 15 degree intervals near the rim of the turntable. This retraction immediately trips a microswitch which starts the drive motor. The turntable is driven 15 degrees, whereupon the spring loaded plunger slips into the next slot, tripping a microswitch which turns off the drive motor. This rotation of 15 degrees is accomplished in about one half second. Thus the operator merely activates the pushbutton switch for an instant whenever he wishes to effect an increment of rotation of the grating.

The grating itself is contructed from nylon monofilament of 0.015 inch diameter spaced at 0.030 inch intervals (thus the width of the grating openings is equal to the diameter of the monofilament, causing half the light to be absorbed by the grating). This generates about N = 115 openings over the 3.5-inch aperture of the grating. A Xerox print of the grating is provided by Figure 5. For the adopted design, the theory of diffraction gratings shows that one fourth of the light impinging on the grating will be directed to the central image; each of the first order images will receive about 10 percent of the original light (or, more precisely, a fraction of $1/\pi^2$ of the light). The angular distance between the central image and each of the first order images is $\theta = \lambda/d$ where λ is the wavelength of the light and d is the center to center spacing of the grating divisions. For blue light ($\lambda = 0.0005$mm) and a spacing of d = 0.030 inches (or 0.75mm), the value of θ becomes 1/1500 radian. Because the MOTS focal length is nominally 1000mm, this corresponds to a spacing of about 0.670mm on the MOTS plate.

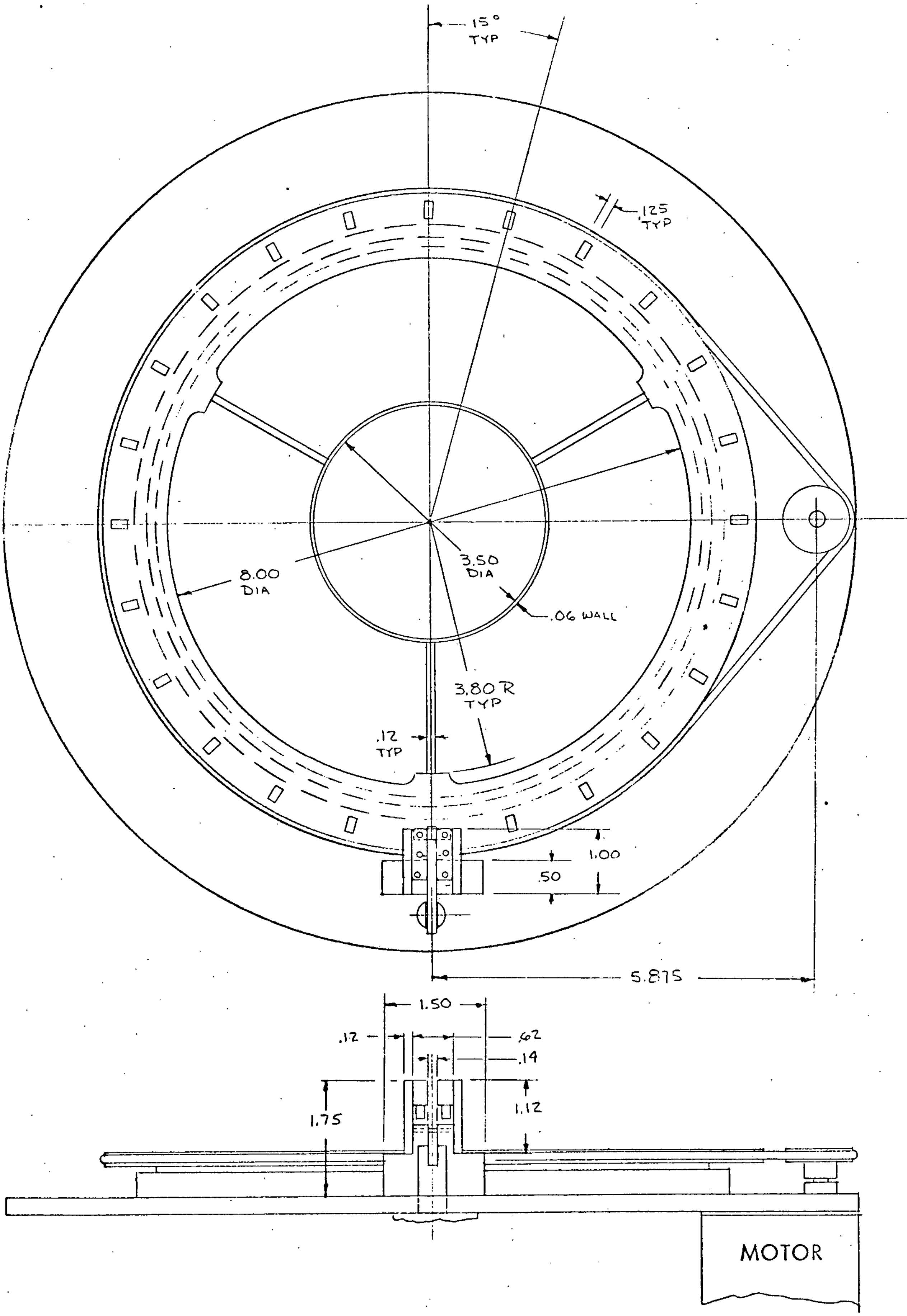


FIGURE 4. Half scale drawing of diffraction grating apparatus employed to monitor MOTS stability.

Event	Time	
Start Drive	$T = 1^h 55^m$ EST	×
Open Shutter	$T + 0^m 59^s$	×
Shift Grating	$T + 1^m 15^s$	×
Shift Grating	$T + 1^m 30^s$	×
Shift Grating	$T + 1^m 45^s$	×
Shift Grating	$T + 2^m 00^s$	×
Shift Grating	$T + 2^m 15^s$	×
Shift Grating	$T + 2^m 30^s$	×
Shift Grating	$T + 2^m 45^s$	×
Shift Grating	$T + 3^m 00^s$	×
Shift Grating	$T + 3^m 15^s$	×
Shift Grating	$T + 3^m 30^s$	×
Shift Grating	$T + 3^m 45^s$	×
Close Shutter & Stop Drive	$T + 4^m 00^s$	×

Event	Time	
Start Drive	$T + 5^m 00^s$	×
Open Shutter	$T + 5^m 59^s$	×
Shift Grating	$T + 6^m 15^s$	×
Shift Grating	$T + 6^m 30^s$	×
Shift Grating	$T + 6^m 45^s$	×
Shift Grating	$T + 7^m 00^s$	×
Shift Grating	$T + 7^m 15^s$	×
Shift Grating	$T + 7^m 30^s$	×
Shift Grating	$T + 7^m 45^s$	×
Shift Grating	$T + 8^m 00^s$	×
Shift Grating	$T + 8^m 15^s$	×
Shift Grating	$T + 8^m 30^s$	×
Shift Grating	$T + 8^m 45^s$	×
Close Shutter & Stop Drive	$T + 9^m 00^s$	×

Event	Time	
Start Drive	$T + 10^m 00^s$	×
Open Shutter	$T + 10^m 59^s$	×
Shift Grating	$T + 11^m 15^s$	×
Shift Grating	$T + 11^m 30^s$	×
Shift Grating	$T + 11^m 45^s$	×
Shift Grating	$T + 12^m 00^s$	×
Shift Grating	$T + 12^m 15^s$	×
Shift Grating	$T + 12^m 30^s$	×
Shift Grating	$T + 12^m 45^s$	×
Shift Grating	$T + 13^m 00^s$	×
Shift Grating	$T + 13^m 15^s$	×
Shift Grating	$T + 13^m 30^s$	×
Shift Grating	$T + 13^m 45^s$	×
Close Shutter & Stop Drive	$T + 14^m 00^s$	×

Event	Time	
Start Drive	$T + 15^m 00^s$	×
Open Shutter	$T + 15^m 59^s$	×
Shift Grating	$T + 16^m 15^s$	×
Shift Grating	$T + 16^m 30^s$	×
Shift Grating	$T + 16^m 45^s$	×
Shift Grating	$T + 17^m 00^s$	×
Shift Grating	$T + 17^m 15^s$	×
Shift Grating	$T + 17^m 30^s$	×
Shift Grating	$T + 17^m 45^s$	×
Shift Grating	$T + 18^m 00^s$	×
Shift Grating	$T + 18^m 15^s$	×
Shift Grating	$T + 18^m 30^s$	×
Shift Grating	$T + 18^m 45^s$	×
Close Shutter & Stop Drive	$T + 19^m 00^s$	×

Event	Time	
Start Drive	$T + 20^m 00^s$	×
Open Shutter	$T + 20^m 59^s$	×
Shift Grating	$T + 21^m 15^s$	×
Shift Grating	$T + 21^m 30^s$	×
Shift Grating	$T + 21^m 45^s$	×
Shift Grating	$T + 22^m 00^s$	×
Shift Grating	$T + 22^m 15^s$	×
Shift Grating	$T + 22^m 30^s$	×
Shift Grating	$T + 22^m 45^s$	×
Shift Grating	$T + 23^m 00^s$	×
Shift Grating	$T + 23^m 15^s$	×
Shift Grating	$T + 23^m 30^s$	×
Shift Grating	$T + 23^m 45^s$	×
Close Shutter & Stop Drive	$T + 24^m 00^s$	×

Event	Time	
Start Drive	$T + 25^m 00^s$	×
Open Shutter	$T + 25^m 59^s$	×
Shift Grating	$T + 26^m 15^s$	×
Shift Grating	$T + 26^m 30^s$	×
Shift Grating	$T + 26^m 45^s$	×
Shift Grating	$T + 27^m 00^s$	×
Shift Grating	$T + 27^m 15^s$	×
Shift Grating	$T + 27^m 30^s$	×
Shift Grating	$T + 27^m 45^s$	×
Shift Grating	$T + 28^m 00^s$	×
Shift Grating	$T + 28^m 15^s$	×
Shift Grating	$T + 28^m 30^s$	×
Shift Grating	$T + 28^m 45^s$	×
Close Shutter & Stop Drive	$T + 29^m 00^s$	×

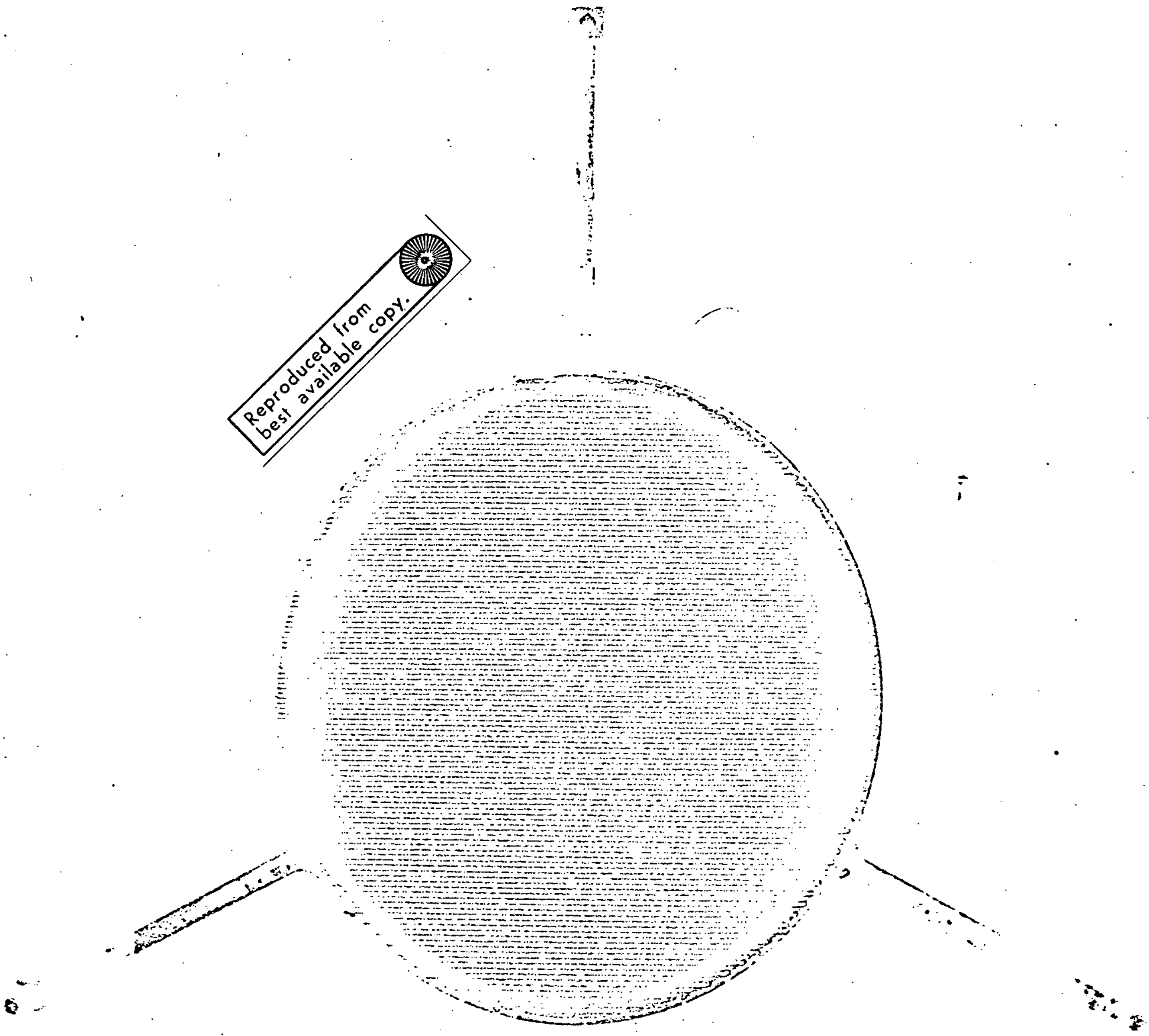


Figure 5. Direct Xerox print of diffraction grating employed in apparatus of Figure 4.

1.4 EXPERIMENTAL PROCEDURE

Short visits were made to the MOTS camera at the Ft. Myers STADAN station in April, September and November 1970 to gather data needed for the evolving design of the apparatus just described. Finally, on the night of December 2, a highly successful series of observations was made with the final version of the diffraction grating. In this section and the next we shall review the results of this test.

Because of other tests being conducted at the same time, the MOTS cameras had been refocussed for optimum imagery with blue light. For this reason, an orthochromatic emulsion (103 Ao) was employed for the diffraction grating experiment. For the data plate that was actually reduced, the MOTS was pointed toward the star β Persius, the drive was started at $01^h\ 55^m$ EST (December 3), and the initial exposure was started one minute later. Thereafter, exposures were made in accordance with the pre-established log reproduced on the next page. The total exposure for a complete set of 12 successive grating positions amounted to 180 seconds (or to two complete periods of the drive error), the grating being shifted at 15 second intervals throughout the exposure. Thus each central image of a star received the full 180 second exposure (not only from the grating but also from the unobscured aperture of the objective, as well), while each pair of diffraction images received a 15 second exposure (from the grating only). To simulate the taking of multiple plates, this process was repeated a total of six times, the drive being turned off for one minute between each sequence. Although all six sequences were recorded on a common plate, the practical end result was equivalent to what would have been obtained from the exposure of a single sequence on each of six separate plates. For this reason, we shall hereafter refer to the results from exposure sequences, 1 through 6 as being from plates 1 through 6.

The star β Persius toward which the camera was initially aimed, generated diffraction images that were too large for precise measurement. However, usable diffraction images were generated by eight other stars ranging in stellar magnitude from 2.5 to 4.0. Inasmuch as a set of diffraction images from a single star is sufficient for monitoring stability, there was ample data from which to make a selection for reduction. In order to obtain a measure of redundancy, we selected two stars from each 'plate' for measurement and reduction. A 40X enlargement of the diffraction images generated by one of the two selected stars is provided in Figure 6. The total exposure of the central image is almost 600 times greater than the exposure of a given diffraction image. This explains the relatively large diameter of the centrl image. Although somewhat ragged because of their small size (about 20 to 30 microns on the plate), the diffraction images are, nonetheless, of a satisfactory quality for precise measurement.

1.5 EXPERIMENTAL RESULTS

On each of the 6 'plates', the 24 diffraction images (12 pairs) for each of the two selected stars were measured by an experienced operator on DBA's Mann comparator. The first phase of the data reduction consisted simply of determining the mean of the coordinates of each of the 12 pairs of diffraction images and then of subtracting from these the grand mean obtained by averaging the set of 12 means. If there were no drift of the camera and no measuring error, the departures from the grand mean would be precisely zero for each averaged pair of diffraction images. Significant, systematic departures were found to exist, and, within acceptable measuring tolerances, were found to be the same for the two stars measured on each plate. Accordingly, the departures for the two stars measured on each plate were themselves averaged for corresponding pairs of diffraction images.

The first stage in the analysis of the results consisted merely of plotting as functions of time the x (right ascension) and y (declination) departures obtained by the process just described. In addition, a program was written to perform a least squares regression based on functions of the form:

$$\delta x_j = a_0 + a_1 \tau_j + a_2 \sin \frac{2\pi}{90} \tau_j + a_3 \cos \frac{2\pi}{90} \tau_j$$

$$\delta y_j = b_0 + b_1 \tau_j + b_2 \sin \frac{2\pi}{90} \tau_j + b_3 \cos \frac{2\pi}{90} \tau_j$$

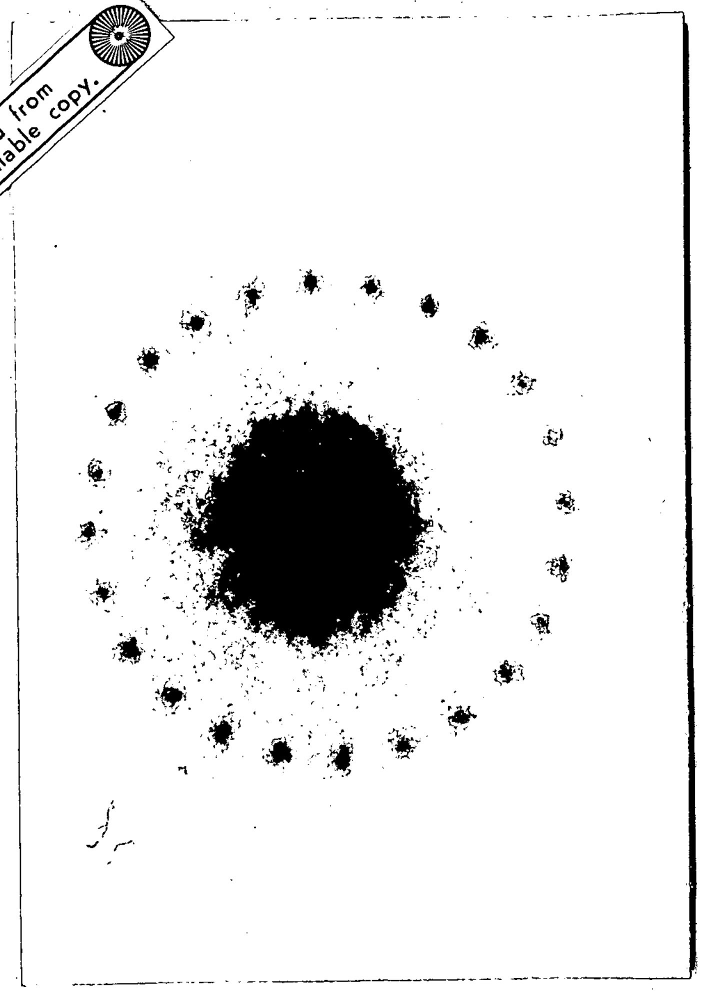


Figure 6. Enlargement (40X) of star exposed through diffraction grating apparatus mounted on MOTS (15 second exposure for each set of diffraction images and 180 second cumulative exposure for central image).

in which,

δx_j , δy_j = observed departures in x and y from grand mean of averaged coordinates for j th pair of diffraction images,

τ_j = relative time of midpoint of exposure of j th pair of diffraction images

= 15 (j - 1) second (i.e., $\tau_1 = 0$, $\tau_2 = 15, \ldots, \tau_{12} = 165$ seconds),

a_0, b_0 = zero set coefficients,

a_1, b_1 = linear drift coefficients,

a_2, a_3
b_2, b_3 = periodic drift coefficients (90 second period assumed).

Values of the observed δx_j , δy_j plotted against τ_j are presented in Figures 7 through 12. Also plotted in the same figures are corresponding values of δx_j , δy_j as computed from the fitted functions.

In reviewing Figures 7 through 12, we find that the departures in right ascension and declination are not randomly distributed about zero, but rather, for the most part, display fairly well defined trends. However, with a few notable exceptions, the fit of the regression functions can be said to be only fair. This is especially clear from the results presented in Table 1 on the following page. These show that goodness of fit indicated by S_x and S_y is generally only modestly better than the dispersion about zero means as indicated by s_x and s_y. Indeed, in a few instances, values of S_x and S_y are slightly larger than their counterparts s_x and s_y, a result attributable to the fewer degrees of freedom associated with the residuals from the fitted functions (i.e., 8 degrees of freedom are associated with S_x, S_y, whereas 11 are associated with s_x, s_y).

It will be noted that the values of s_x and S_x for plate 6 are exceptionally large (9.1 and 7.8μm, respectively). Referring to the plotted results in Figure 12, we see the presence of two pronounced jumps in the trends as indicated by the heavy arrows. To make sure that these jumps were actually real and not the result of a measuring or recording

Table 1. RMS values of x and y drifts before and after regression.

Plate	RMS Values Before Regression (μm)		RMS Values After Regression (μm)	
	s_x	s_y	S_x	S_y
1	3.4	2.5	2.4	1.8
2	3.5	2.3	3.4	1.8
3	4.0	1.7	3.7	1.9
4	5.3	2.3	4.3	1.1
5	3.2	2.1	3.3	1.9
6	9.1	2.3	7.8	2.2

$$s_x = \left[\frac{\Sigma\, \delta x_j^2}{11}\right]^{\frac{1}{2}} \qquad S_x = \left[\frac{\Sigma\, (\delta x_j - \delta x'_j)^2}{8}\right]^{\frac{1}{2}}$$

$$s_y = \left[\frac{\Sigma\, \delta y_j^2}{11}\right]^{\frac{1}{2}} \qquad S_y = \left[\frac{\Sigma (\delta y_j - \delta y'_j)^2}{8}\right]^{\frac{1}{2}}$$

δx_j, δy_j = observed values

$\delta x'_j$, $\delta y'_j$ = values computed from regression

PLATE NO. 1

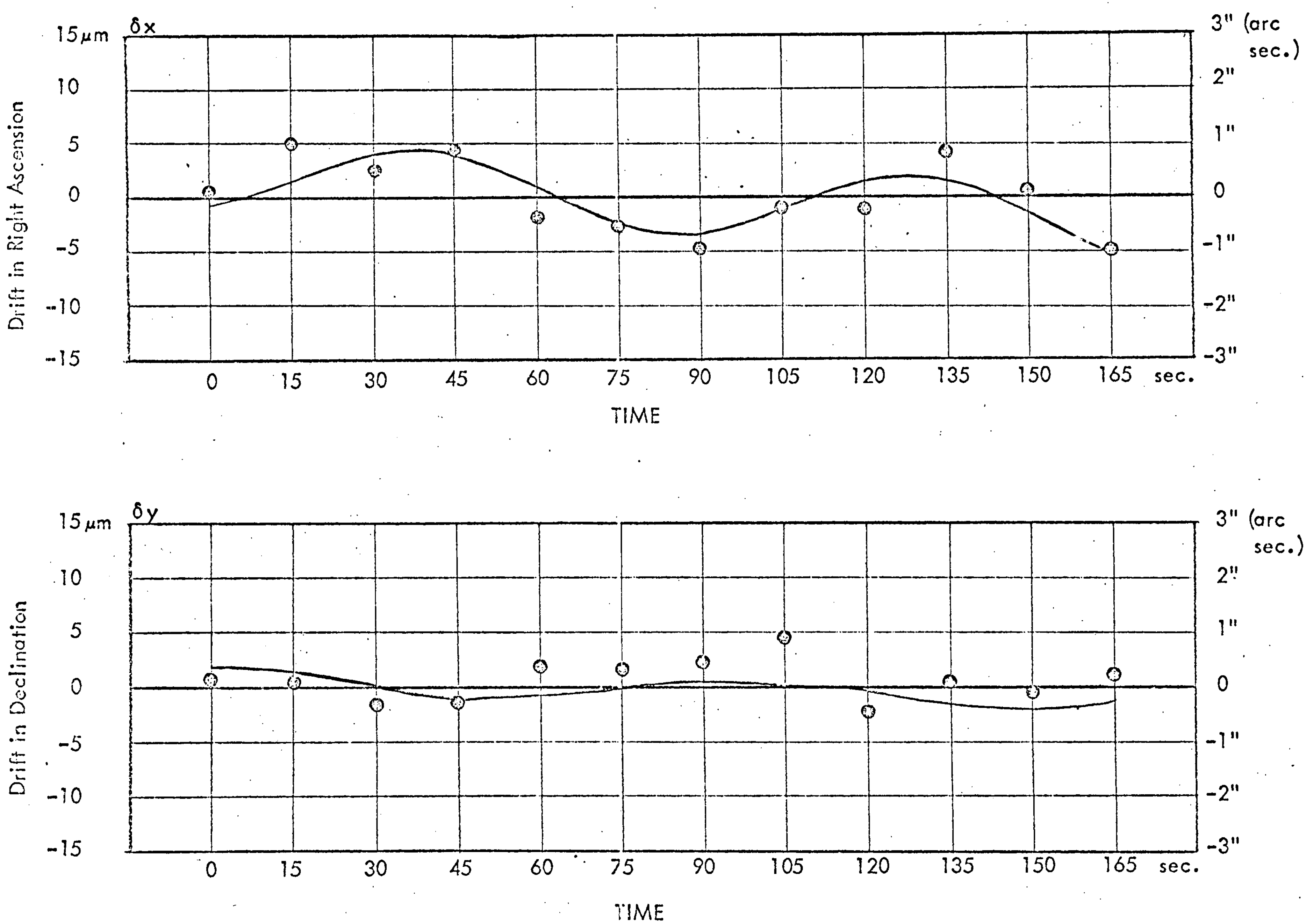


Figure 7 . MOTS drift as indicated by rotating diffraction grating, plate no. 1 .

PLATE NO. 2

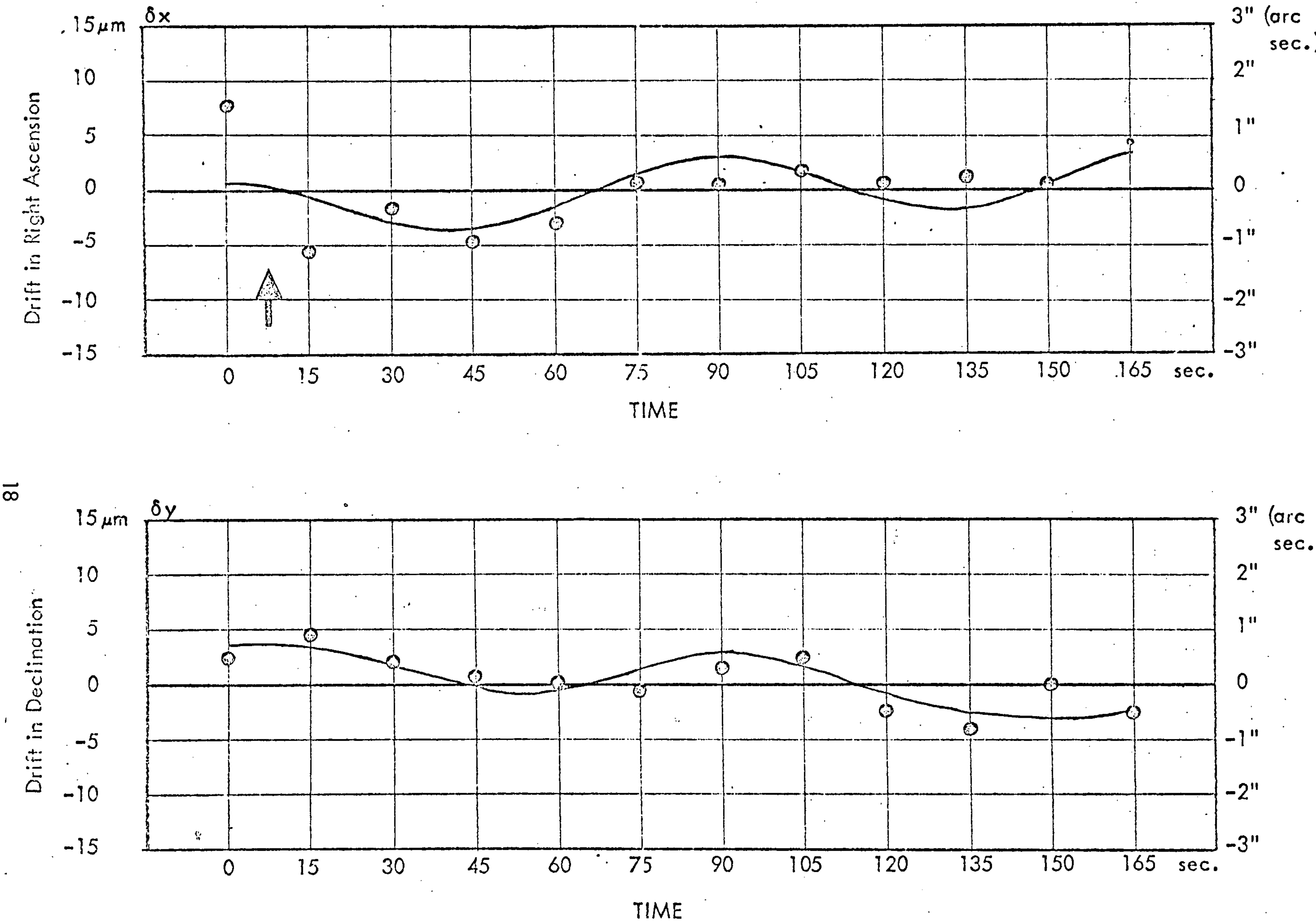


Figure 8 . MOTS drift as indicated by rotating diffraction grating, plate no. 2 .

PLATE NO. 3

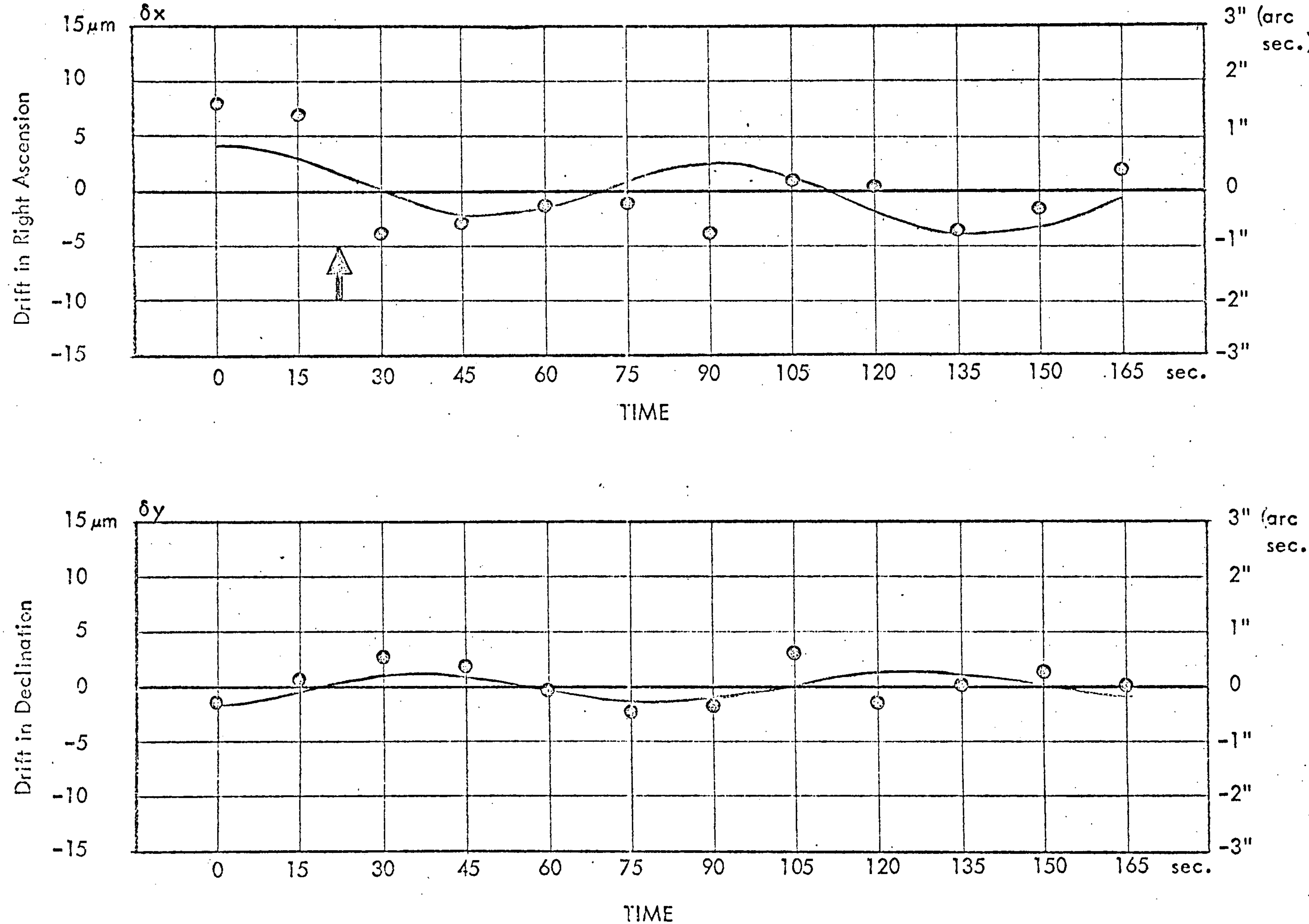


Figure 9 . MOTS drift as indicated by rotating diffraction grating, plate no. 3 .

PLATE NO. 4

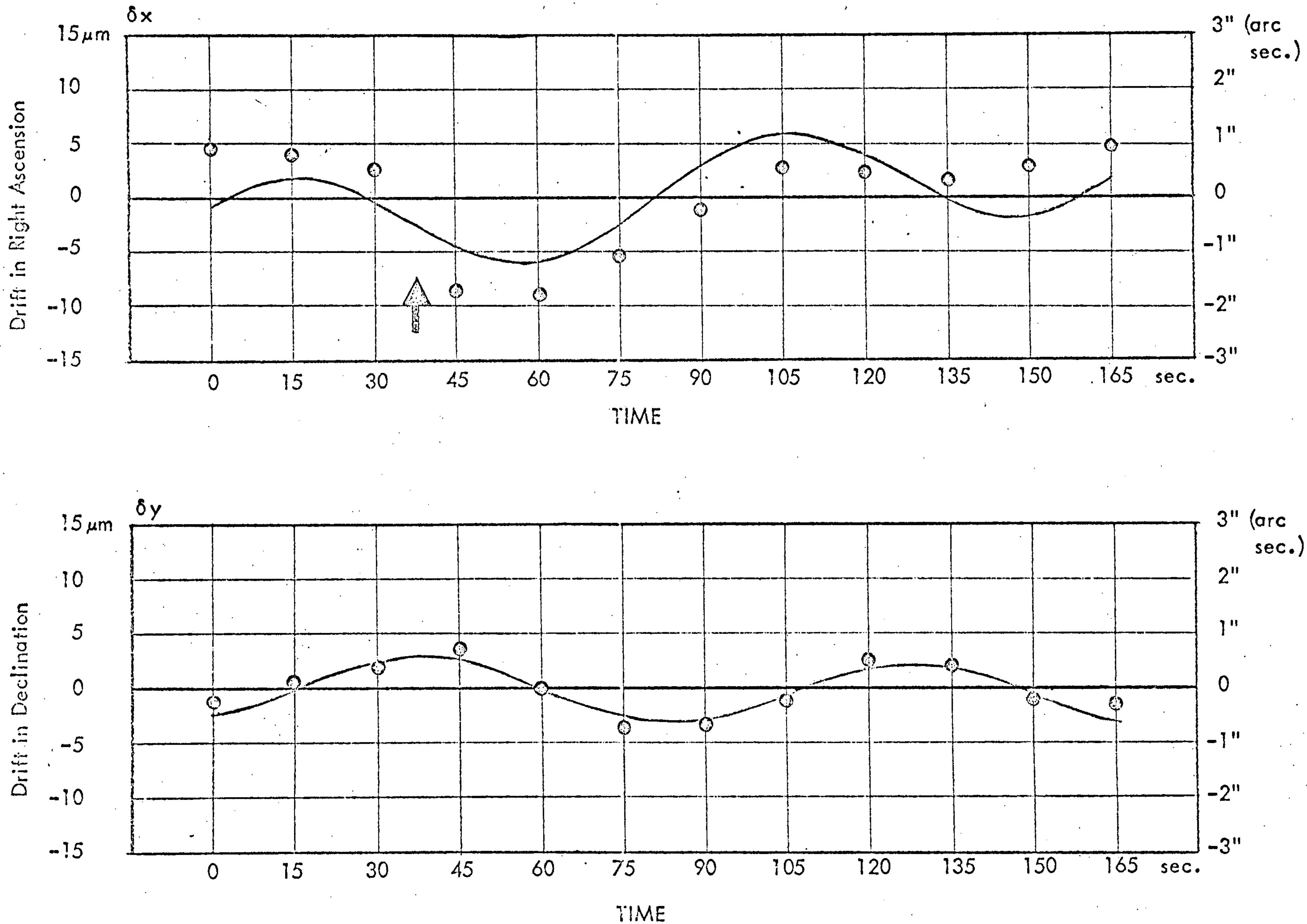


Figure 10. MOTS drift as indicated by rotating diffraction grating, plate no. 4.

PLATE NO. 5

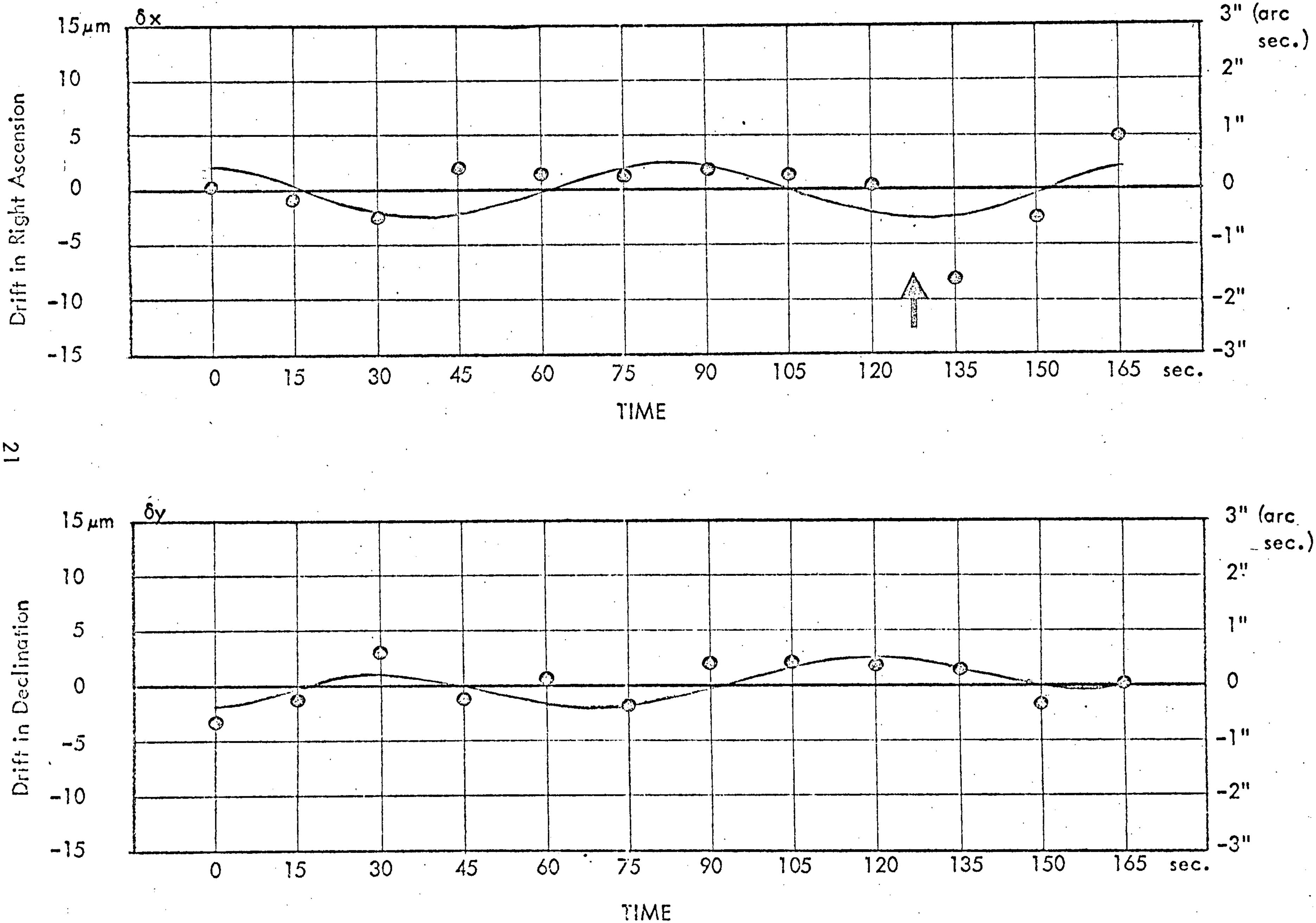


Figure 11 . MOTS drift as indicated by rotating diffraction grating, plate no. 5 .

PLATE NO. 6

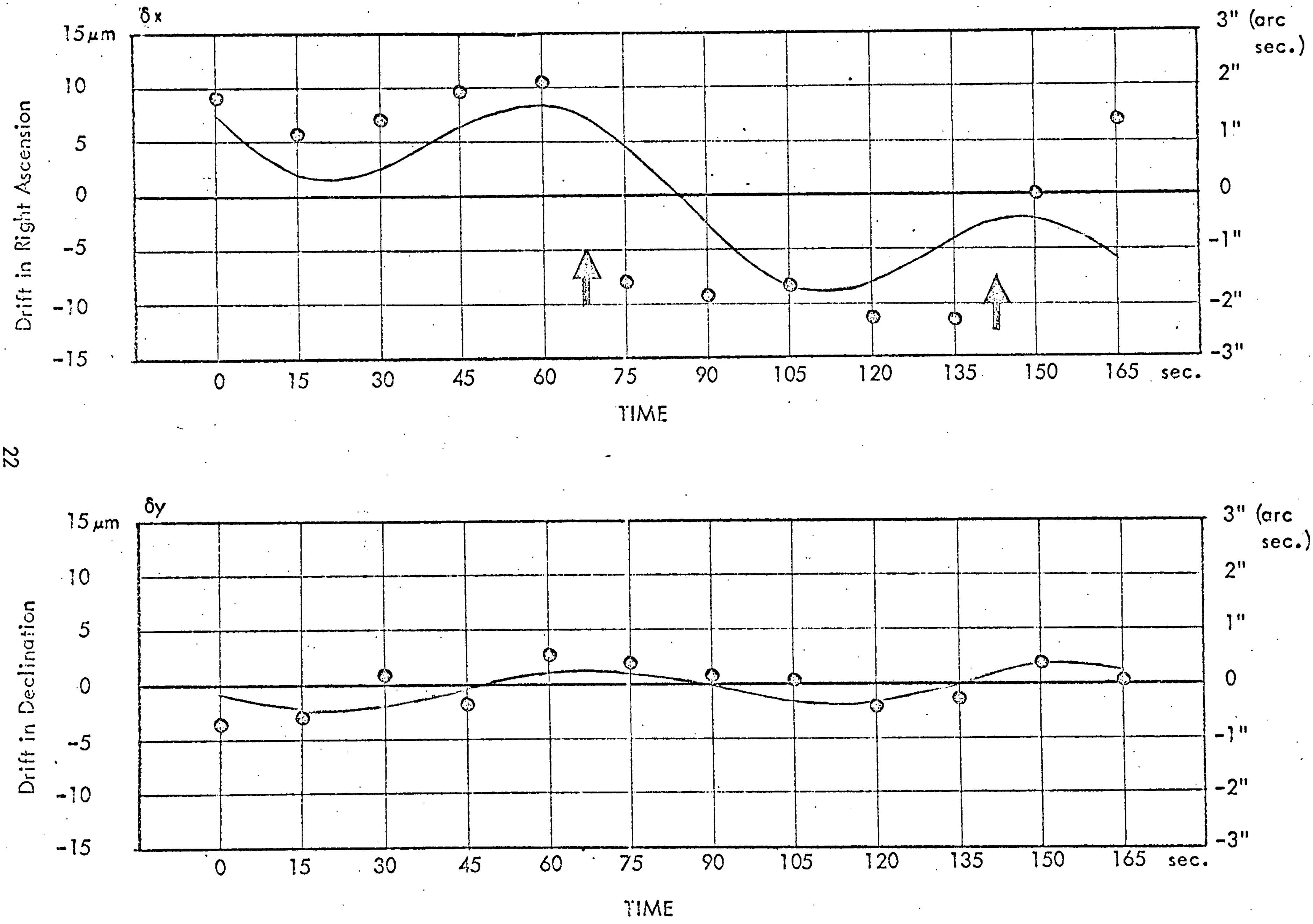


Figure 12. MOTS drift as indicated by rotating diffraction grating, plate no. 6 .

PLATE NO. 6

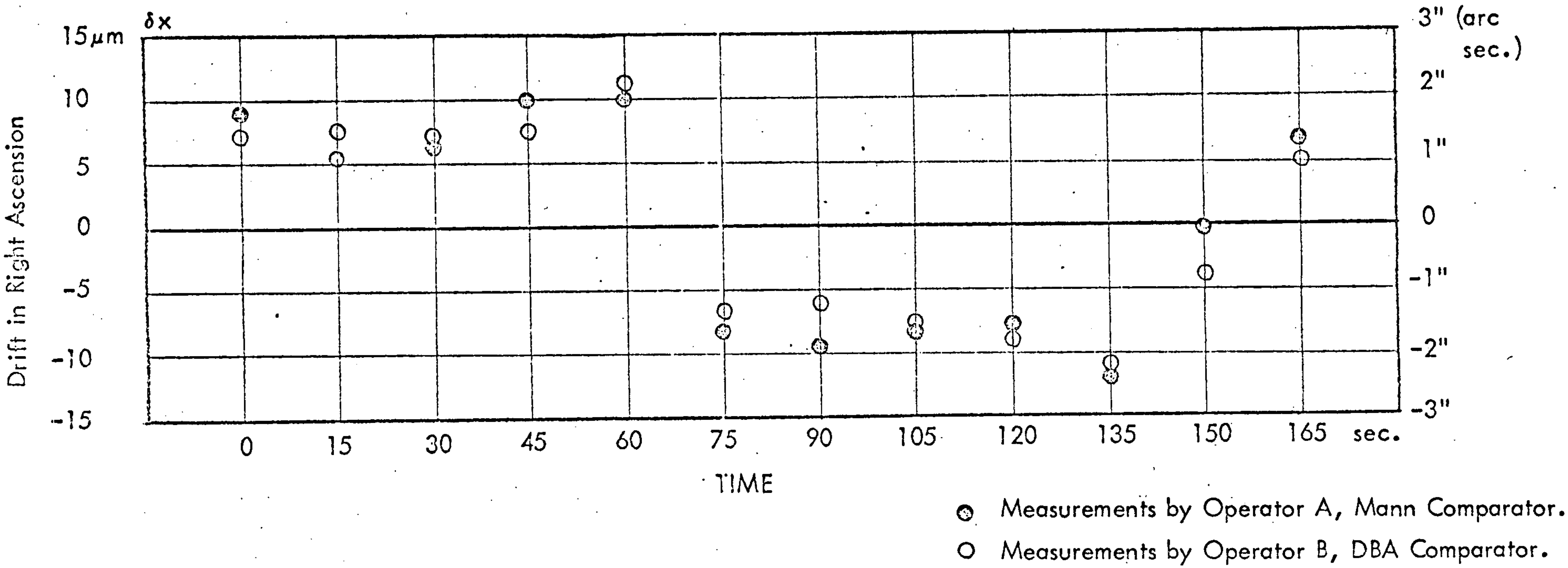


Figure 13. Comparison of original measurements of drift in right ascension for plate 6 with remeasurements made on different comparator by different operator.

blunder, we had plate 6 remeasured by another operator on another comparator (namely, the DBA multilaterative comparator). Comparative results presented in Figure 13 show good agreement between the two sets of measurements and verify that the jumps are indeed real. Reviewing the plots for the other plates, we find indications of similar jumps occurring on Plates 2, 3, 4 and 5 at the points indicated by the arrows in Figures 8, 9, 10 and 11. These suggest the presence of stiction in the drive train. Stiction being the occasional slight binding of the gears followed by sudden release. Whether or not this is the correct explanation for the abrupt changes remains to be determined. For convenience, we shall nonetheless refer to the abrupt changes as 'stiction jumps'.

It seemed that the generally mediocre fits obtained from the regressions performed on δx could well be attributable to the compromising effect of stiction jumps. To test this hypothesis we revised the regression model so that the single zero set term a_0 was replaced by the expression:

$$\xi_1 a_0 + \xi_2 a_0' + \xi_3 a_0''$$

in which,

$$\begin{aligned} \xi_1 &= 1 \text{ for } \tau_1 < T_1 \\ &= 0 \text{ for } \tau_1 > T_1 \\ \xi_2 &= 1 \text{ for } T_1 < \tau_1 < T_2 \\ &= 0 \text{ otherwise} \\ \xi_3 &= 1 \text{ for } \tau_1 > T_2 \\ &= 0 \text{ otherwise.} \end{aligned}$$

This artifice makes it possible to reinitialize the zero set term up to two times per plate. When no reinitializations are desired, we need merely set T_1 and T_2 to values greater than the maximum value of τ_1 (say to 180 seconds). The values of T_1 and T_2 adopted in the revised regressions for δx are as indicated in the following table.

Plate	Number of Stiction Jumps	T_1	T_2	Degrees of Freedom For Regression
1	0	180.	180	8
2	1	7.5	180	7
3	1	22.5	180	7
4	1	37.5	180	7
5	1	127.5	180	7
6	2	67.5	142.5	6

The results of the revised regressions are plotted in Figurcs 14 through 19. The results for Plate 1 are unchanged, which is as it should be, since no stiction jumps were exercised. Results for the remaining plates represent substantial improvements over those from the original regressions. The fitted functions now very closely follow the observed values. This is verified by Table 2 in which Table 1 is extended to reflect the results of the revised regressions. The grand rms value for the fit in δx for all six plates is 2.1 μm, a value only insignificantly greater than the grand rms of 1.8 μm for y.

The amplitude of the periodic error from the revised regressions averages about 0.6 second of arc in right ascension and about 0.4 second of arc in declination. This is appreciably lower than the values considered typical (i.e. 2 to 3 arc seconds in right ascension) in Reference 2. However, specific results from the MOTS at Fort Myers were not reported in Reference 2.

1.6 GENERAL CONSIDERATIONS

When stiction jumps are duly taken into account whenever they occur, excellent fits are obtained from the simple regression model that was adopted. This demonstrates the effectiveness of the rotating diffraction grating as a means for monitoring the stability of the MOTS camera throughout routine operations.

Because of the exploratory nature of our investigation, we took the liberty of ignoring a few fine points that should be implemented when the diffraction data are actually to be used to generate corrections applicable to satellite observations. Because

Table 2. Extension of Table 1 to include results of revised regressions.

Plate	RMS Values Before Regression (μm)		RMS Values After Original Regression (μm)		RMS Values After Revised Regression (μm)	
	s_x	s_y	S_x	S_y	S'_x	S'_y
1	3.4	2.5	2.4	1.8	2.4	1.8
2	3.5	2.3	3.4	1.8	1.5	1.8
3	4.0	1.7	3.7	1.9	2.0	1.9
4	5.3	2.3	4.3	1.1	1.4	1.1
5	3.2	2.1	3.3	1.9	2.8	1.9
6	9.1	2.3	7.8	2.2	2.0	2.2

$$s_x = \left[\frac{\Sigma\, \delta x_j^2}{11}\right]^{\frac{1}{2}} \qquad S_x = \left[\frac{\Sigma\,(\delta x_j - \delta x'_j)^2}{8}\right]^{\frac{1}{2}} \qquad S'_x = \left[\frac{\Sigma\,(\delta x_j - \delta x''_j)^2}{f}\right]^{\frac{1}{2}}$$

$$s_y = \left[\frac{\Sigma\, \delta y_j^2}{11}\right]^{\frac{1}{2}} \qquad S_y = \left[\frac{\Sigma\,(\delta y_j - \delta y'_j)^2}{8}\right]^{\frac{1}{2}} \qquad S'_y = \text{same as } S_y$$

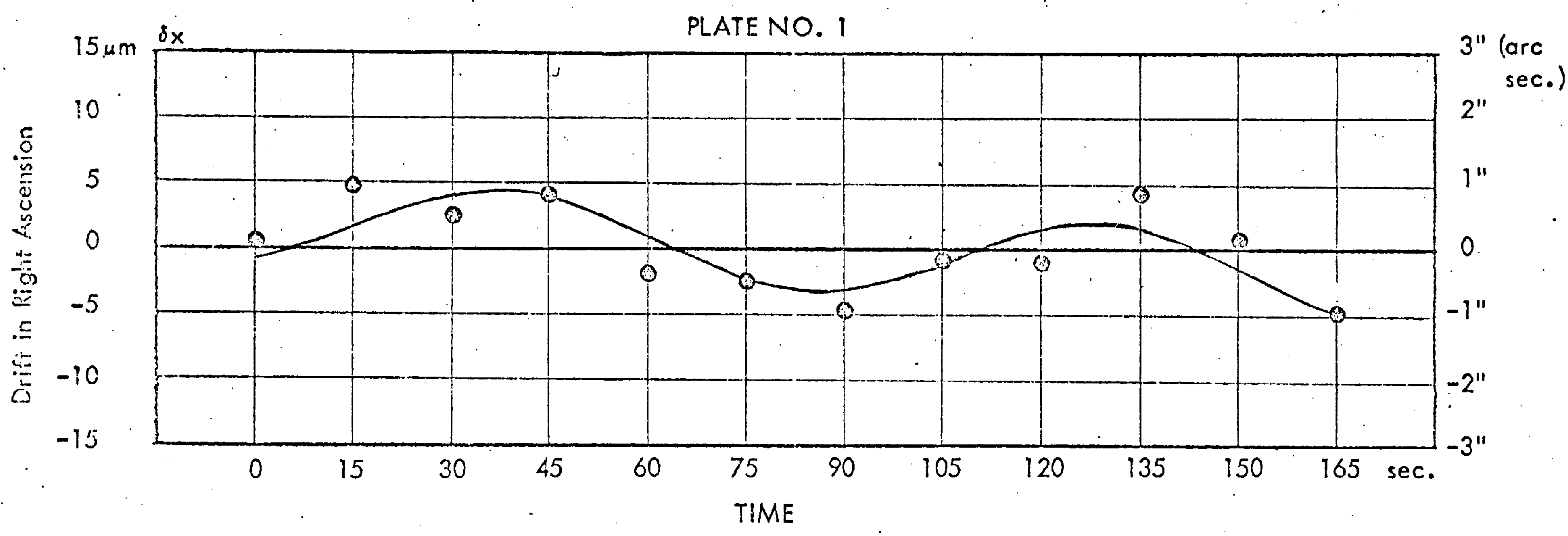
$\delta x_j, \delta y_j$ = observed values

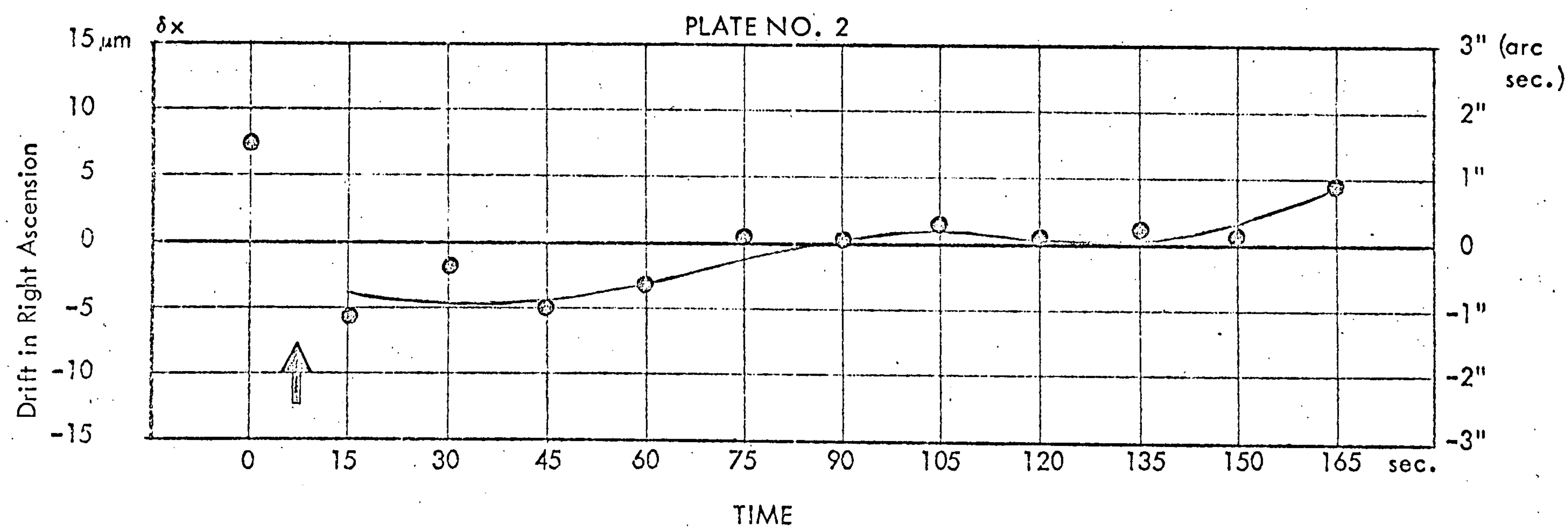
$\delta x'_j, \delta y'_j$ = values computed from original regression

$\delta x''_j, \delta y''_j$ = values computed from revised regression

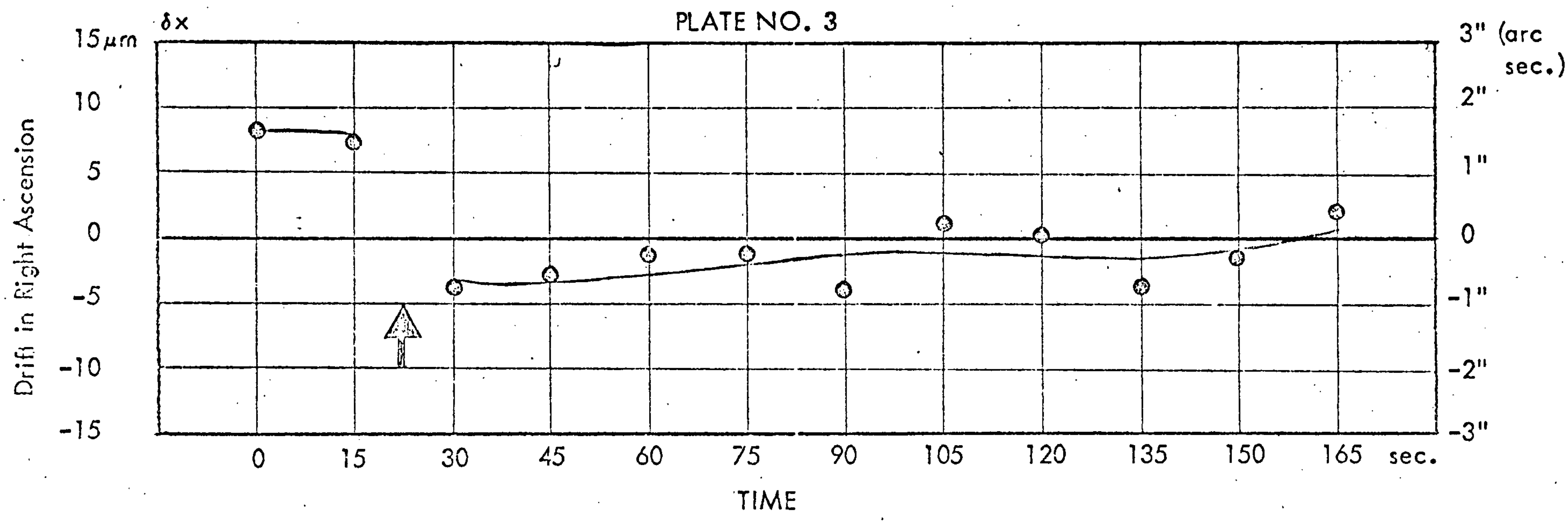
f = degrees of freedom

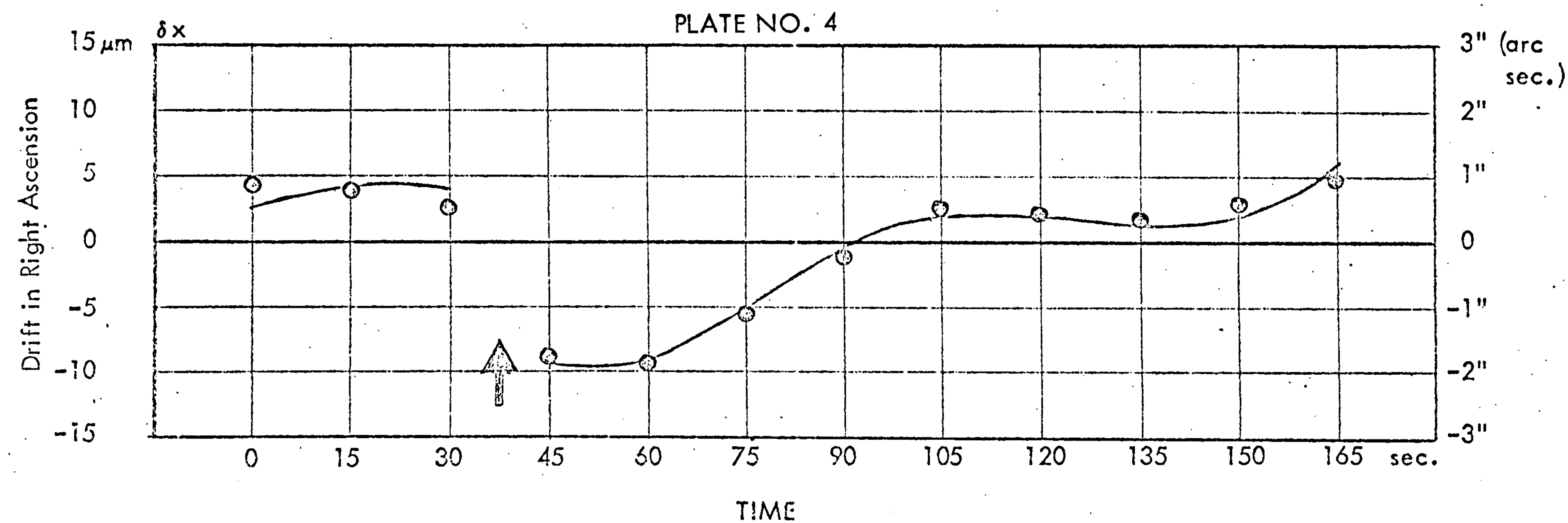
= 8, 7 or 6 depending on number of stiction jumps



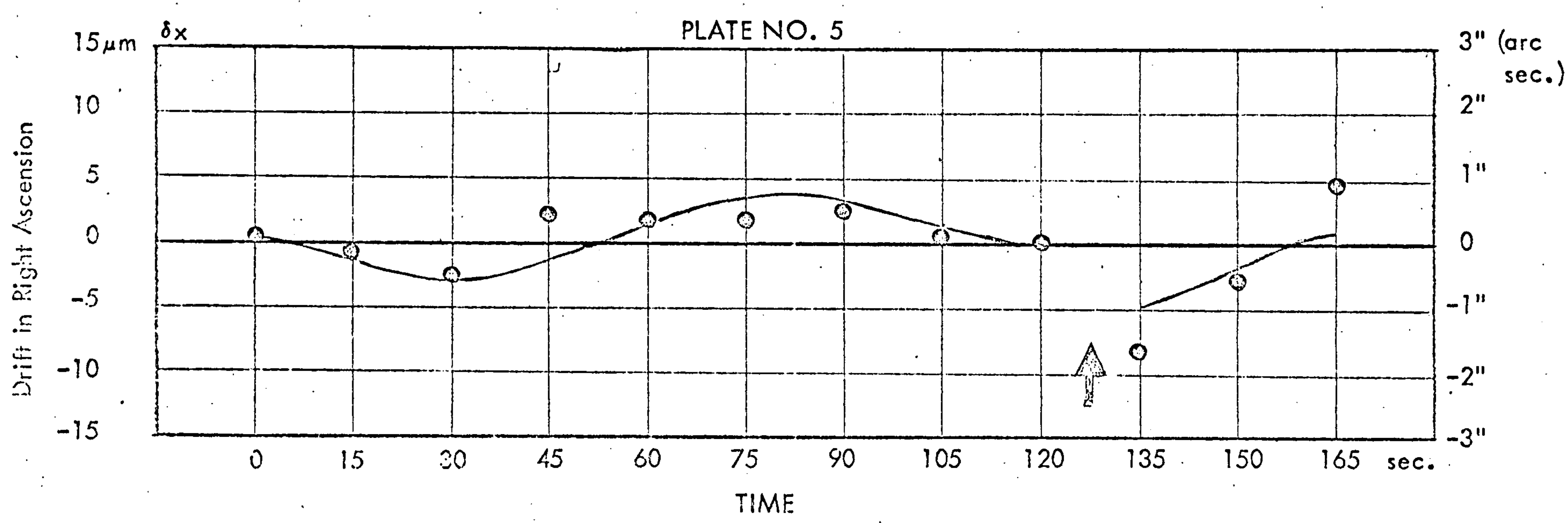


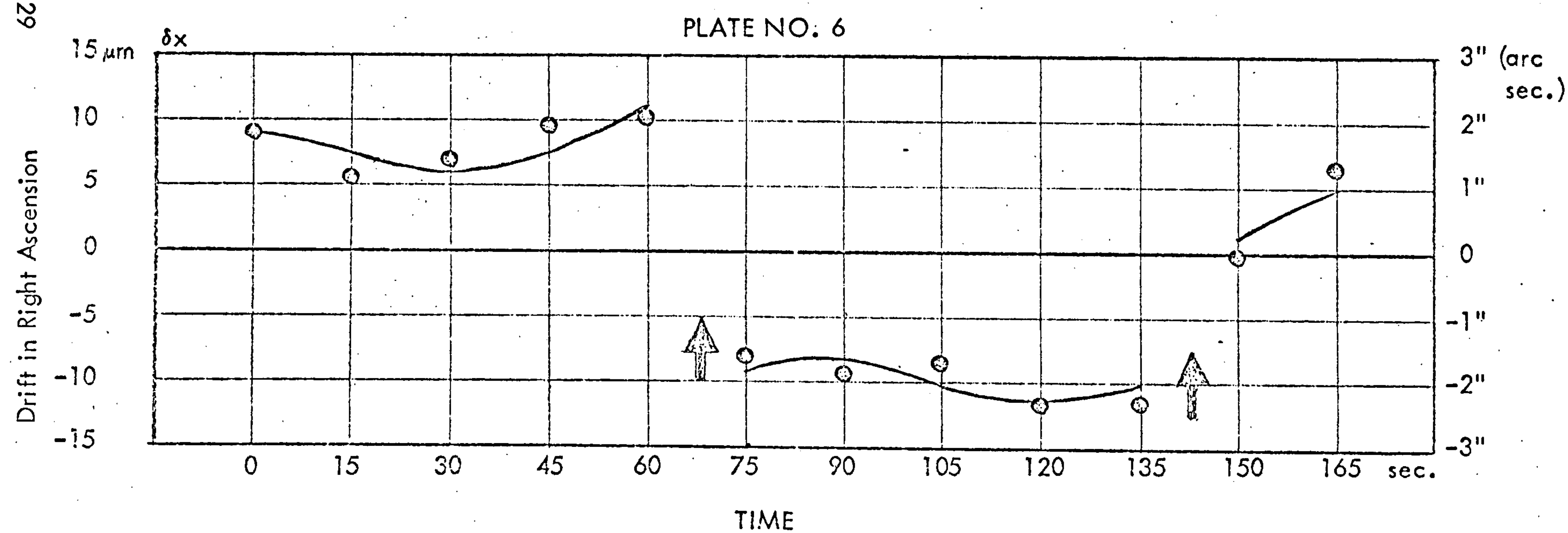
Figures 14 and 15. Plots showing fit from revised regressions in right ascension, plates 1 and 2.





Figures 18 and 19. Plots showing fit from revised regressions in declination, plates 3 and 4.





Figures 16 and 17. Plots showing fit from revised regressions in right ascension, plates 5 and 6.

the length of each exposure through the grating may be a significant fraction of the 90 second period of the drive, the integrated effect of the drift over exposure increments should be recognized as being what is actually observed. Accordingly, if 2Δ denotes the length of each exposure through the grating, the expression

$$a_2 \sin \frac{2\pi}{90} \tau_1 + a_3 \cos \frac{2\pi}{90} \tau_1$$

in the formula for δx_1, for example, should more properly be replaced by the expression

$$\frac{1}{2\Delta} \int_{\tau_1 - \Delta}^{\tau_1 + \Delta} [a_2 \sin \frac{2\pi}{90} t + a_3 \cos \frac{2\pi}{90} t] \, dt .$$

However, this reduces to the form

$$a_2' \sin \frac{2\pi}{90} \tau_1 + a_3' \cos \frac{2\pi}{90} \tau_1$$

in which

$$a_2' = K a_2$$

$$a_3' = K a_3$$

where

$$K = \left(\sin \frac{2\pi}{90} \Delta\right) \Big/ \left(\frac{2\pi}{90} \Delta\right).$$

Thus, while a_2' and a_3' are the values actually obtained from the least squares regression, the values to be used in the correction formulas should be a_2, a_3 which will, of course, be slightly larger. For $2\Delta = 15$ seconds (the value used in our experimental investigation), the value of K is $(\sin \pi/6)/(\pi/6) = 0.956$, which means that the amplitudes obtained from the regressions should be increased by a factor of $1/.956 = 1.044$.

The second fine point to be considered in operational applications of the diffraction grating concerns effects induced by changes in atmospheric refraction resulting from the gradual changes in zenith distance occurring throughout the operation. Even if the MOTS were perfectly stable in the right ascension-declination frame, the gradual change in refraction with change in zenith distance would be manifested as a secular drift if appropriate corrections were not applied. In Reference 3 we showed that the correction for this effect is given by:

$$\underset{\text{(arc sec)}}{d(\Delta\zeta)} = 4.5 \times 10^{-3}\, t_i \sin A_i \cos \Phi \sec^2 \zeta_i$$

in which, for the present application,

$d(\Delta\zeta)$ = angular correction to be applied zenith distance of mean of pair of diffraction images

t_i = time of exposure of i th set of diffraction images relative to mean time of total exposure

A_i = azimuth of star at t_i

ζ_i = zenith distance of star at t_i

Φ = latitude of station.

By means of standard astronomical formulas, this correction for zenith distance can be propagated into right ascension and declination and thence into x,y plate coordinates. It is the corrected x,y coordinates that properly should be employed in the diffraction grating regressions.

1.7 CONCLUSIONS AND RECOMMENDATIONS

The diffraction grating apparatus evolved from this study has proven to provide a simple, inexpensive, effective, and operationally unobtrusive means for monitoring the stability of the MOTS camera to a precision of a few tenths of a second of arc.

We recommend that NASA employ the breadboard apparatus developed by DBA in further tests on other MOTS cameras, particularly those that are known to be affected by larger drift errors than the MOTS at Ft. Myers. Should these tests confirm the promise of the approach, NASA should undertake the procurement of operationally optimized units to be used routinely in future operations at all MOTS stations. Such units should preferably be controlled by an automatic programmer with a suitable range of selectable exposure rates.

While the diffraction grating method provides a way of correcting for drift in future MOTS operations, the question naturally arises as to whether anything can be done about the many hundreds of plates gathered on past operations, particularly on GEOS I and GEOS II. We believe that quite possibly something effective can be done. Because a formal standardized procedure was followed in the exposure of GEOS plates, a fairly high level of repeatibility of drive error may well exist. In any event, whether or not this is the case can be determined experimentally by employing the grating in repeated trials simulating GEOS operations. If acceptably repetitive results are obtained for a given camera, a pooled result can be used to derive corrections to be applied to directions obtained from previous plates taken by the camera. The validity of such corrections can be tested by determining whether or not improved residuals are generally obtained from short arc reductions based on revised directions.

1.8 REFERENCES

[1] Brown, D., "Advanced Methods for the Calibration of Metric Cameras," final report, Contract No.: DA-44-009-AMC-1457(X), U.S. Army Engineering Topographic Laboratories, Fort Belvoir, Virginia, 9 December 1968.

[2] Harris, D., Cartwright, M., Oosterhaut, J., "Analysis of the MOTS Camera Drive," GSFC Report X-514-69-482, November 1969.

[3] Brown, D., Hartwell, J., Stephenson, J., "Geodetic Data Analysis for GEOS A, An Experimental Design," final report prepared for NASA Goddard under Contract No. NAS5-9860, November 1965.

SECTION 2

MINITRACK DATA REDUCTION

2.1 INTRODUCTION

This section covers work done by DBA Systems, Inc., over a 2-year period. Work for the contract was initially being performed by the DBA Florida Office. In order to provide better liaison betweem NASA and DBA, it was decided to transfer the project to the DBA Washington Office. This was done nine months after the start of the contract. The section dealing with the Minitrack geometrical error model, contributed by James B. Willmann, was written prior to the transfer of the project to Washington. Several thousand NAP (Network Analysis Program-II) control cards for processing optical orbits had also been prepared, but this work became largely redundant after the development of a new optical preprocessing program.

After the project was transferred to Washington, work was concentrated on calibrating Minitrack using short optically determined reference arcs. This was done in order to demonstrate the feasibility of using the NAP program for the purpose of calibrating Minitrack. At this time too, the magnitude of the Doppler effect was being investigated, and the relevant equations were implemented into the NAP program.

Concurrently, the new optical preprocessing program, already mentioned, was developed. The reason for writing a new optical preprocessing program was that the old program could not process data for SAO stations and required an enormous amount of data preparation. The new program in addition would punch out most of the NAP control cards.

Twelve months after the start of the contract, work began on the calibration of Minitrack using Minitrack data only. Initially, little progress was made. At this time the Minitrack extract and sort programs were developed. The Minitrack preprocessor was modified to punch out most of the NAP control cards. After three months, it was found that the NAP program had been requested to compute and apply the effect of the earth's precession and nutation. Unfortunately, an error was known to exist in these computations. After the NAP control cards were changed, so as not to request these computations, progress was again made. (To the order of the accuracy of the Minitrack system, it would not appear to be necessary to consider precession and nutation. However, because of the surprisingly good agreement between orbits determined from Minitrack and optical data, this point of view may have to be modified).

Definite but slow progress was now being made. A post NAP program -- later extensively modified -- was written to summarize the results of the NAP program. This also involved modifications to NAP subroutines RESID and FINALP.

About 17 months after the start of the contract, the contract Technical Monitor (W. M. Rice) proposed an analysis of the NAP program by Boole & Babbage, Inc., to determine the feasibility (within the time scale of the contract) of increasing the computational speed of NAP and hence the overall rate of progress. The analysis showed that the majority of the computing time was being spent in a few relatively short subroutines. A new algorithm was then developed for the computation of spherical harmonics and this was implemented into NAP. The modifications to the NAP program doubled its overall computational speed.

Advantage was now taken of the faster NAP program, the length of arc considered was increased from 2.5 to 5 days; however, this led to further problems. A meeting was called by E. P. Damon of the Computer Systems Branch to call attention to the amount of computer disk space being required by the NAP program. (As the length of arc had been doubled, so had the amount of data being stored on disk memory). We had, however, become aware of the problem before the above mentioned meeting because of the frequently aborted runs due to non-availability of enough disk space. To overcome this problem, the special subroutines PRTIAL and RESID were written and the NAP disk space requirements were substantially reduced The post-NAP program was modified to be consistent with the latest NAP program.

With the increasing amount of data being processed on each run, the number of NAP control cards became very large and it was decided to write them on tape. To handle modifications to individual NAP "cards" the pre-NAP card updater program was written.

In checking out the new special subroutine PRTIAL a discrepancy was discerned in the printed out time. This was traced back to the Minitrack preprocessing program, which was corrected. All Minitrack data used in the data processing now had to be reprocessed.

At about 21 months from the start of the contract it was decided to investigate the puzzlingly slow rate of convergence of the solutions computed by NAP. The investigation revealed that the NAP matrix inversion was not being handled as described in the NAP documentation but in an iterative manner, which was extremely time consuming for the long arcs being considered. The complete equations as given in the existing NAP documentation were incorporated into NAP by R. Garza-Robles of Goddard Space Flight Center (Code 551). The rate of convergence was at least doubled.

With the help of the support programs already described and with the modified NAP program a four-arc run was now made on the 360/95. The total time span covered was 25 days and the computer time taken was 20 minutes. (At the start of this program, one 2.5 day arc was processed in 24 minutes and then the convergence rate of the solution was half of its present rate). "Optical" reference arcs were computed for the same time spans and a comparison of results was made. Advantage was taken of the current DBA contract (NAS5-11730) to develop an ionospheric error model for incorporation in the NAP program and one run was made using ionospheric corrections.

It should also be mentioned that the NAP-II program was successfully overlaid reducing its core requirements from 700,000 to 500,000 bytes.

This section is devoted to discussion of the reduction of Minitrack data using the NAP-II program and auxiliary programs written specifically for this task. Included in this section are discussions of the Minitrack error model and the Doppler effect and wave-propagation time delay as applied to predicted Minitrack measurements.

The second and third parts of this section discuss the handling and preprocessing of the Minitrack and Optical data respectively. The fourth part is devoted to the reduction of the data using the NAP-II program.

Results for a reduction effort are given in the fifth part. Results are given for the station at Ft. Myers and then a multi-station network which includes all of the Minitrack stations.

The last part of this section is a recommended procedure for reducing Minitrack data on a production basis.

As of special note: Appendix A-4 should be continuously referenced in the data processing cycle, as it contains restrictions on the programs and assumptions pertaining to the data.

2.1.1 The Minitrack Geometrical Error Model

This discussion will consider some of the general geometric properties of the Minitrack system. As with any interferometer system, the basic measurement mode is that of the phase path difference between the target and two antennas located some distance apart. The line between the antennas is referred to as the base line. The geometry for two cases is depicted below; 'A' for a target at a close range, and 'B' for a target at infinity.

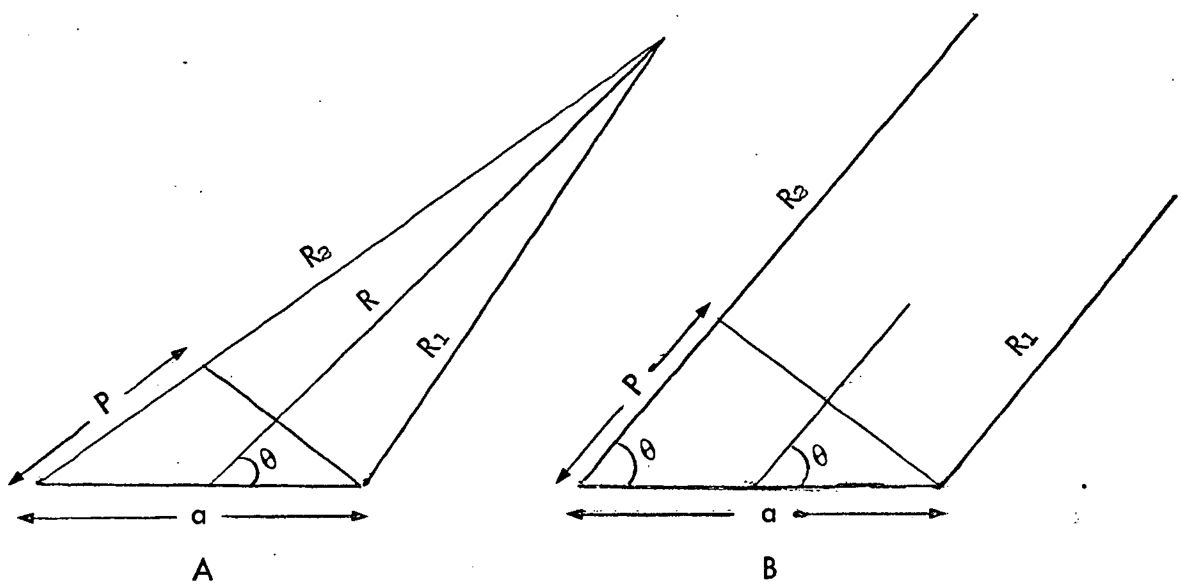


Figure 1

In Figure (1), A and B, the antennas are at the end of the base line a. The measurement made by the system can be interpreted as being proportional to the distance P. In both cases, P is defined as

$$P = R_2 - R_1 .$$

Generally, we say that the system measures the cosine of the angle θ since

$$\cos\theta \approx \frac{P}{a} .$$

For the case of a point at infinity, the above expression is exact. For the case depicted in Figure 1A, it is only approximately true and $\cos\theta$ depends upon R as well as P and a.

For the Minitrack system, there are four antennas located at the ends of two perpendicular base lines. These base lines are usually located to intersect each other at their centers. However, since exact location of the phase centers of the antennas is not possible, there will be errors in the resulting data if they are interpreted as a true measurement of the cosine of the direction to the target. Therefore, the discussion which follows will investigate two areas. First, we will derive the relationship between the angle θ and the measurement P. Then, we will investigate errors in the antenna locations and how they effect the interpretation of the data.

To derive the desired expression, we can refer to Figure 1A. From this figure, the following relationships can be written.

$$P = R_2 - R_1$$

$$R_1^2 = R^2 + \left(\frac{a}{2}\right)^2 - R a \cos\theta$$

$$R_2^2 = R^2 + \left(\frac{a}{2}\right)^2 + R a \cos\theta \quad .$$

We will redefine the measurement as

$$\ell_o = \frac{P}{a} = \frac{R_2 - R_1}{a} \quad .$$

Then ℓ_o approximates the cosine of θ. Proceeding with the algebra, give

$$P^2 + 2PR_1 + R_1^2 = R_2^2$$

$$P^2 + 2PR_1 - 2Ra\cos\theta = 0$$

$$(P^2 - 2Ra\cos\theta)^2 = 4P^2 R_1^2 = 4P^2\left(R^2 + \frac{a^2}{4} - Ra\cos\theta\right)$$

$$P^4 + 4R^2 a^2 \cos^2\theta = 4P^2 R^2 + P^2 a^2$$

solving for $\cos\theta$ give

$$\cos\theta = \ell_o \sqrt{1 + \frac{a^2}{4R^2}\left(1 - \ell_o^2\right)}$$

the desired result. Note that as $R \to \infty$ then $\ell_0 \to \cos\theta$. Expanding gives

$$\cos\theta = \ell_0 + \frac{a^2}{8R^2}\ell_0(1 - \ell_0^2) \ldots$$

The maximum error in assuming $\cos\theta = \ell_0$ occurs when ℓ_0 is approximately $1/\sqrt{3}$. This leads to a maximum error of about

$$\cos\theta - \ell_0 \simeq \frac{a^2}{20R^2} \quad .$$

For this analysis, we will consider the three dimensional case pictured in Figure (2). The antennas are located along the x and y base lines which are approximately a and b in length. We will then relate the path difference measurements to the desired cosines as a function of a, b and the errors in antenna location. This is pictured below.

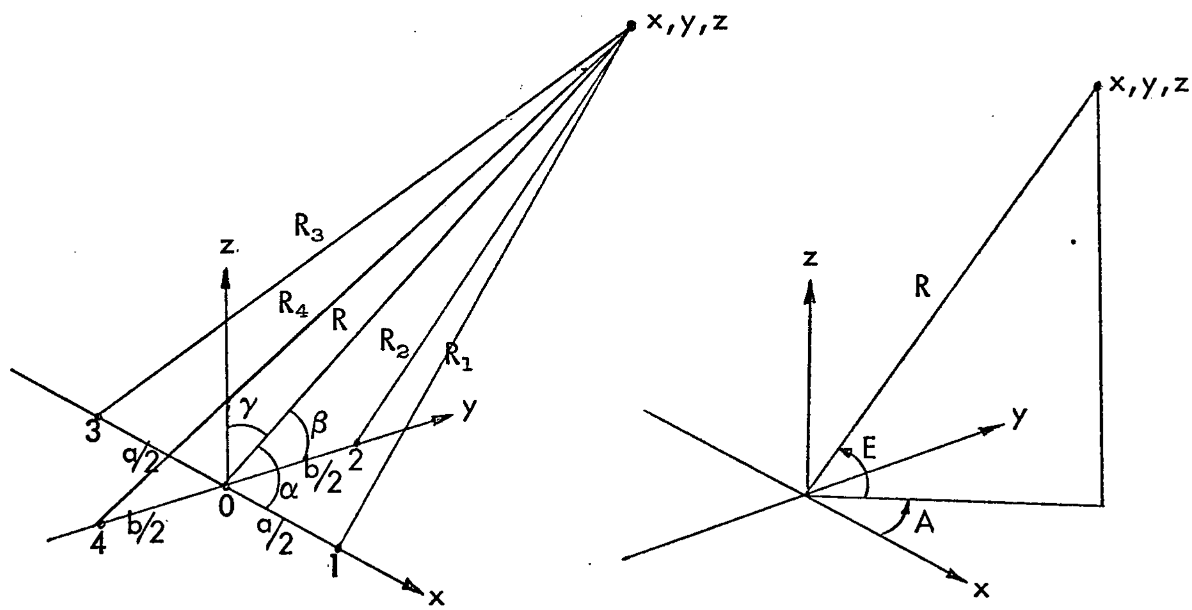


Figure 2

We have also included a figure which defines the measurements of azimuth and elevation. First a few definitions.

$$\ell = \cos\alpha$$

$$m = \cos\beta$$

$$n = \cos\gamma = \sqrt{1 - \cos^2\alpha - \cos^2\beta} = \sqrt{1 - \ell^2 - m^2}$$

$$x = R\cdot\ell = R\cdot\cos A\cos E$$

$$y = R\cdot m = R\cdot\sin A\cos E$$

$$z = R\cdot n = R\cdot\sin E\ .$$

Thus,

$$\tan A = m/\ell\ ; \quad \cos E = \sqrt{\ell^2 + m^2}\ .$$

The measurements made by the system are given by

$$P = R_3 - R_1 = \sqrt{(x + \frac{a}{2} - x_3)^2 + (y - y_3)^2 + (z - z_3)^2}$$

$$- \sqrt{(x - \frac{a}{2} - x_1)^2 + (y - y_1)^2 + (z - z_1)^2}$$

$$Q = R_4 - R_1 = \sqrt{(x - x_4)^2 + (y + \frac{b}{2} - y_4)^2 + (z - z_4)^2}$$

$$- \sqrt{(x - x_2)^2 + (y - \frac{b}{2} - y_2)^2 + (z - z_2)^2}\ .$$

In the above, we have chosen the coordinates of the antennas to be

1 $x_1 + a/2,\ y_1,\ z_1$

2 $x_2,\ y_2 + b/2,\ z_2$

3 $x_3 - a/2,\ y_3,\ z_3$

4 $x_4,\ y_4 - b/2,\ z_4$.

In the case where there are no errors, then $x_1, y_1, \ldots, y_4, z_4$ would all be zero. In that case, if P_o is the value of P when all survey errors are zero, then

$$P_o = \sqrt{(x + a/2)^2 + y^2 + z^2} - \sqrt{(x - a/2)^2 + y^2 + z^2}$$

$$= \sqrt{R^2 + ax + a^2/4} - \sqrt{R^2 - ax + a^2/4}$$

and

$$Q_o = \sqrt{R^2 + by + b^2/4} - \sqrt{R^2 - by + b^2/4} \quad .$$

The solution to these two equations is the same as that presented in Section II. That is

$$\ell = \frac{P_o}{a}\sqrt{1 + \frac{a^2}{4R^2}\left(1 - \frac{P_o^2}{a^2}\right)} = \frac{P_o}{a} + \frac{a^2}{8R^2} \quad \frac{P_o}{a}\left(1 - \frac{P_o^2}{a^2}\right) + \ldots$$

$$m = \frac{Q_o}{b}\sqrt{1 + \frac{b^2}{4R^2}\left(1 - \frac{Q_o^2}{b^2}\right)} = \frac{Q_o}{b} + \frac{b^2}{8R^2} \quad \frac{Q_o}{b}\left(1 - \frac{Q_o^2}{b^2}\right) + \ldots$$

However, of more importance is the effect on ℓ and m when the errors in antenna location are not zero. To explore this, we can expand the equations for P and Q about $x_1 \ldots z_4 = 0$. This gives

$$P = P_o - \frac{(x + a/2)\, x_3 + yy_3 + zz_3}{R_3} + \frac{(x - a/2)\, x_1 + yy_1 + zz_1}{R_1} \quad .$$

For purposes of this expansion, we can let $R_3 = R_1 = R$, giving

$$P = P_o + \frac{x}{R}(x_1 - x_3) + \frac{y}{R}(y_1 - y_3) + \frac{z}{R}(z_1 - z_3) - \frac{a}{2R}(x_1 + x_3)$$

but

$$\ell = \frac{x}{R}, \quad m = \frac{y}{R}, \quad n = \frac{z}{R}$$

thus,

$$P = P_o + \ell(x_1 - x_3) + m(y_1 - y_3) + n(z_1 - z_3) - \frac{a}{2R}(x_1 + x_3) \quad .$$

The actual measurement is given by

$$\ell_o = \frac{P}{a} ,$$

and from before

$$\frac{P_o}{a} \simeq \ell - \frac{a^2}{8R^2}\ell(1 - \ell^2) \approx \ell \; .$$

Therefore, we finally have

$$\ell = \ell_o - \ell_o \frac{(x_1 - x_3)}{a} - m_o \frac{(y_1 - y_3)}{a} - n_o \frac{(z_1 - z_3)}{a} + \frac{(x_1 + x_3)}{2R} \quad .$$

Proceeding in the same manner, we have for m

$$m = m_o - \ell_o \frac{(x_2 - x_4)}{b} - m_o \frac{(y_2 - y_4)}{b} - n_o \frac{(z_2 - z_4)}{b} + \frac{y_2 + y_4}{2R} .$$

Normally, the last term of both expressions can be neglected. Alternately, if a and b are chosen so that $x_1 + x_3$ and $y_2 + y_4$ are zero, then it can be dropped from consideration.

For close-in ranges when the approximation

$$\ell = \frac{P_o}{a}$$

cannot be made, then the value of ℓ and m above can be used to compute the exact cosine value through the solution of

$$\ell = \frac{P_o}{a} + \frac{a^2}{8R^2} \frac{P_o^2}{a^2} \left(1 - \frac{P_o^2}{a^2}\right) .$$

2.1.2 The Doppler Effect and Wave-Propagation Time-Delay as Applied to Predicted Minitrack Measurements

Consider a satellite at position $\underline{r}(t)$ at time t transmitting an unmodulated signal at a constant frequency ν. The signal is received by two MINITRACK antennas situated at $\frac{a}{2}\underline{i}$ and $-\frac{a}{2}\underline{i}$, respectively, where a is the baseline length and $\underline{i}$ is a unit vector in the x-direction. The phase difference between two signals arriving at the two antennas at the same time t is measured. If the two signals left the satellite at times t_A and t_B, respectively, then the phase-difference between the two signals is $2\pi\nu(t_A - t_B)$. The actual measurement, $\Delta\varphi$, is the phase-difference divided by 2π. Hence,

$$\Delta\varphi = \nu\,(t_A - t_B)\,. \tag{1}$$

or equivalently,

$$\Delta\varphi = \frac{c\,(t_A - t_B)}{\lambda}\,. \tag{2}$$

where c is the velocity of light and λ the wavelength. We must also have,

$$\left|\underline{r}_A - \frac{a}{2}\underline{i}\right| = c(t - t_A)\,, \tag{3}$$

and

$$\left|\underline{r}_B + \frac{a}{2}\underline{i}\right| = c(t - t_B)\,, \tag{4}$$

where, in general, $\underline{r}_1 = \underline{r}\,(t_1)$

Writing,

$$\tau = t_A - t_B\,, \tag{5}$$

we have by the mean-value theorem

$$\underline{t}_B = \underline{t}_A - \tau\,\dot{\underline{r}}_A + \tfrac{1}{2}\tau^2\,\ddot{\underline{r}}_M\,, \tag{6}$$

where t_M is a time between t_A and t_B.

Writing

$$r_A = |\underline{r}_A| \text{ and } \underline{p}_A = \underline{r}_A / r_A , \tag{7}$$

we obtain from (6),

$$\underline{r}_B + \tfrac{1}{2} a \underline{i} = r_A \left[\underline{p}_A - \left(\frac{a}{r_A}\right)\left(\frac{c\tau}{a}\right) \frac{\dot{\underline{r}}_A}{c} + \tfrac{1}{2}\left(\frac{c\tau}{a}\right)^2 \left(\frac{a}{r_A}\right) \frac{a\ddot{\underline{r}}_M}{c^2} + \tfrac{1}{2}\left(\frac{a}{r_A}\right)\underline{i} \right] \tag{8}$$

Since $\underline{p}_A{}^2 = 1$, $\underline{p}_A \cdot \dot{\underline{r}}_A = \dot{r}_A$, and $\underline{i} \cdot \underline{p}_A = l_A$,

the direction cosine at time t_A , we obtain from (8),

$$|\underline{r}_B + \tfrac{1}{2} a \underline{i}| = r_A \left\{ 1 + \left(\frac{a}{r_A}\right)\left[l_A - 2\, l_0 \frac{\dot{r}_A}{c} + l_0{}^2 \frac{a\, \underline{p}_A \cdot \ddot{\underline{r}}_M}{c^2} \right] + \left(\frac{a}{r_A}\right)^2 \left[\tfrac{1}{2}\underline{i} - l_0 \frac{\dot{\underline{r}}_A}{c} + \tfrac{1}{2} l_0{}^2 \frac{a\ddot{\underline{r}}_M}{c^2} \right]^2 \right\}^{\frac{1}{2}} \tag{9}$$

where $$l_0 = \frac{c\tau}{a} \tag{10}$$

Next we make some assumptions regarding the magnitude of the terms in (9).

We assume that,

$$|l_0| \leqq 1 \tag{11}$$

(This will be justified later on when we shall show that l_0 is a first approximation to the direction cosine l_A)

Write, $$\frac{a}{r_A} = a_3 , \tag{12}$$

and assume that $a_3 < 10^{-3}$, i.e. that the satellite is always more than 1000 baseline lengths from the station.

Write $$\frac{\dot{r}_A}{c} = u_4 \tag{13}$$

and assume that $|\dot{\underline{r}}_A|/c < 10^{-4}$, i.e. that the satellite velocity is less than 10^{-4} times the velocity of light. Since $|\dot{r}_A| \leqq |\dot{\underline{r}}_A|$ it follows that $|u_4| < 10^{-4}$.

Next assume that, $\frac{r}{c^2} \quad |\underline{\ddot{r}}| \; < \; 10^{-7}$ (14)

Hence by the assumption made in (12) ,

$$\frac{a}{c^2} \; |\underline{\ddot{r}}_M| \; < \; 10^{-10} \tag{15}$$

With the above assumptions equation (9) may be rewritten as,

$$|\underline{r}_B + \tfrac{1}{2} a \, \underline{i}\,| = r_A \left\{ 1 + a_3 \left[l_A - 2 l_0 \, u_4 + \frac{1}{4} \, a_3 + 0 \; (10^{-7}) \right] \right\}^{\frac{1}{2}} \tag{16}$$

where $0 \; (10^{-n})$ denotes terms of order 10^{-n} or less.

Equation (16) may be rewritten in the form:

$$|\underline{r}_B + \frac{1}{2} \, a \; \underline{i} \;\, | = r_A \left\{ 1 + a_3 l_A + \frac{1}{4} a_3^2 \, l_A^2 + a_3 \left[\frac{1}{4} \, a_3 \; (1 - l_A^2) - 2 l_0 \, u_4 + 0 \; (10^{-7}) \right] \right\}^{\frac{1}{2}}$$

$$= r_A (1 + \frac{1}{2} \, a_3 l_A) \left\{ 1 + a_3 \left[\frac{1}{4} \, a_3 \, (1 - l_A^2) - 2 \, l_0 u_4 + 0(10^{-7}) \right] \left[1 - a_3 l_A + 0(10^{-6}) \right] \right\}^{\frac{1}{2}}$$

$$= r_A (1 + \frac{1}{2} \, a_3 l_A) \left\{ 1 + \frac{1}{2} \, a_3 \left[\frac{1}{4} \, a_3 (1 - l_A^2) - 2 \, l_0 \, u_4 \; + 0(10^{-7}) \right] \left[1 - a_3 l_A \right] \right\}$$

$$= r_A \left\{ 1 + \frac{1}{2} a_3 l_A + \frac{1}{2} \, a_3 \left[\frac{1}{4} \, a_3 \, (1 - l_A^2) \, (1 - \frac{1}{2} \, a_3 \, l_A) - 2 \, l_0 \, u_4 + 0(10^{-7}) \right] \right\} \tag{16'}$$

We obtain a similar expression for $|\underline{r}_1 - \frac{1}{2} \, a \; \underline{i}\,|$ with a_3 replaced by $-a_3$ and $u_4 = 0$.

Hence:

$$|r_A - \frac{1}{2} \; a \; \underline{i} \; | = r_A \left\{ 1 - \frac{1}{2} \, a_3 \; l_A + \frac{1}{2} \; a_3 \left[\frac{1}{4} \; a_3 \, (1 - l_A^2) \, (1 + \frac{1}{2} \, a_3 l_A + 0 \; (10^{-7}) \right] \right\} \tag{17}$$

Hence from (3), (4), (5), (16)', and (17),

$$c\tau \; = r_A \, a_3 \left\{ l_A - \frac{1}{8} \; a_3^2 \; (1 - l_A^2) \; l_A - l_0 \, u_4 + 0 \; (10^{-7}) \right\} \tag{18}$$

It is easy to show that $\frac{1}{8} a_3^2 (1 - l_A^2)\, l_A = 0(10^{-7})$. Hence from (10), (12), and (18)

$$l_0 = l_A - l_0 u_4 + 0\,(10^{-7}).$$

Whence, by (13)

$$l_A = l_0 \left(1 + \frac{\dot{r}_A}{c}\right) + 0(10^{-7}) \tag{19}$$

Hence we have shown that l_0 is a first approximation to the direction cosine l_A.

From (2), (5), and (10)

$$l_0 = \frac{\lambda\, \Delta\varphi}{a} , \tag{20}$$

and hence

$$l_A = \frac{\Delta\varphi}{a} \left[\lambda \left(1 + \frac{\dot{r}_A}{c}\right)\right] + 0\,(10^{-7}) \tag{21}$$

From expression (21) it is easy to see that the term in $(\dot{r}_A/c)$ represents the Doppler effect. We shall show that expression (21) may be further simplified.

From (3) and (17) we have

$$c\,(t - t_A) = r_A \left[1 + 0(10^{-3})\right] \tag{22}$$

Hence denoting the direction cosine at time t by l, we have by the mean value theorem

$$l = l_A + \frac{r_A}{c} \left[1 + 0(10^{-3})\right] \dot{l}_A + \tfrac{1}{2} \frac{r_A^2}{c^2} \left[1 + 0(10^{-3})\right] \ddot{l}_N \tag{23}$$

where l_N is the direction cosine at some time t_N between t and t_A. Writing $\underline{p} = \underline{r}/r$ we obtain

$$r\, \dot{\underline{p}} = \dot{\underline{r}} - \dot{r}\, \underline{p}$$

$$= \underline{p} \times (\dot{\underline{r}} \times \underline{p}) \tag{24}$$

since $\underline{p}^2 = 1$ and $\underline{p} \cdot \dot{\underline{r}} = \dot{r}$. Differentiating (24) gives,

$$r\, \ddot{\underline{p}} = \dot{\underline{p}} \times (\dot{\underline{r}} \times \underline{p}) + \underline{p} \times (\ddot{\underline{r}} \times \underline{p}) + \underline{p} \times (\dot{\underline{r}} \times \dot{\underline{p}}) - \dot{r}\, \dot{\underline{p}} \tag{25}$$

Since $I = \underline{p} \cdot \underline{i}$ it follows that,

$$I \leqq |\underline{p}|, \quad \dot{I} \leqq |\dot{\underline{p}}| \text{ and } \ddot{I} \leqq |\ddot{\underline{p}}| \tag{26}$$

By (24),

$$r\,|\dot{\underline{p}}| \leqq |\dot{\underline{r}}| \tag{27}$$

From (25) and (27)

$$\begin{aligned} r^2\,|\ddot{\underline{p}}| &\leqq |\dot{\underline{r}}|^2 + r\,|\ddot{\underline{r}}| + |\dot{\underline{r}}|^2 + \dot{r}\,|\dot{\underline{r}}| \\ &\leqq 3\,|\dot{\underline{r}}|^2 + r\,|\ddot{\underline{r}}| \end{aligned} \tag{28}$$

From (26) and (27) we have

$$\begin{aligned} \frac{r\,\dot{I}}{c} &\leqq \frac{|\dot{\underline{r}}|}{c} \\ &= 0\,(10^{-4}) \end{aligned} \tag{29}$$

by our previous assumptions. From (26) and (28) we obtain

$$\begin{aligned} \tfrac{1}{2}\,\frac{r^2\,\ddot{I}}{c^2} &\leqq \tfrac{1}{2}\left[3\,\frac{|\dot{\underline{r}}|^2}{c^2} + r\,\frac{|\ddot{\underline{r}}|}{c^2}\right] \\ &= 0\,(10^{-7})\,, \end{aligned} \tag{30}$$

by the assumptions made in (13) and (14) above.

We then have from (23), (29), and (30),

$$I = I_A + \frac{r_A}{c}\,\dot{I}_A + 0(10^{-7}) + \frac{r_A^{\,2}}{r_N^{\,2}}\,0\,(10^{-7}) \tag{31}$$

By the mean value theorem

$$r_N = r_A + (t_N - t_A)\,\dot{r}_K$$

But $t_N - t_A \leqq t - t_A = \frac{r_A}{c}\left[1 + 0\,(10^{-3})\right]$, Hence

$$\begin{aligned} |t_N - t_A|\,|\dot{r}_K| &\leqq r_A\,|\dot{r}_K|/c\,\left[1 + 0\,(10^{-3})\right] \\ &= r_A\,0\,(10^{-4}), \text{ where} \end{aligned}$$

$$r_N > r_A - r_A\,0(10^{-4}), \text{ and } (r_A/r_N) \leqq 1 + 0\,(10^{-4})$$

Hence $I = I_A + \frac{r_A}{c}\,\dot{I}_A + 0\,(10^{-7})$ (32)

But $r_A \dot{l}_A = \dot{x}_A - \dot{r}_A l_A$, (33)

where $\dot{x}_A = \underline{\dot{r}}_A \cdot \underline{i}$

Hence $l = l_A + \frac{\dot{x}_A}{c} - l_A \left(\frac{\dot{r}_A}{c}\right) + 0\,(10^{-7})$ (34)

From (19) we obtain

$$l_0 = l_A \left(1 - \frac{\dot{r}_A}{c}\right) + 0\,(10^{-7}) ,$$

Hence from the above and (36)

$$l = l_0 + \frac{\dot{x}_A}{c} + 0\,(10^{-7}) \qquad (35)$$

By the mean value theorem and (22)

$$\frac{\dot{x}}{c} = \frac{\dot{x}_A}{c} + \frac{r_A}{c}\,\frac{\ddot{x}_D}{c}\left[1 + 0\,(10^{-3})\right]$$

$$= \frac{\dot{x}_A}{c} + 0\,(10^{-7}), \text{ by (14)},$$ where $\dot{x}$ is the x-component of the velocity at time t.

Hence,

$$l_0 = l - \frac{\dot{x}}{c} + 0\,(10^{-7}) .$$

Finally from the above and (20),

$$\boxed{\frac{\lambda\,\Delta\varphi}{a} = l - \frac{\dot{x}}{c} + 0\,(10^{-7})} \qquad (38)$$

The following assumptions were made in deriving (38)

$$(a/r) < 10^{-3} \quad , \quad (|\underline{\dot{r}}|/c) < 10^{-4} \quad ,$$

and $r\,|\underline{\ddot{r}}|/c^2 < 10^{-7}$. If the only forces acting on the satellite are due to the Earth's gravity field then $|\underline{\ddot{r}}| = gR_E^2/R^2$, where g is the acceleration due to gravity at the Earth's surface, R_E is the radius of the Earth and R is the radius vector from the center of the Earth to the satellite. Since $r \leqq 2R$ it then follows that $r\,|\underline{\ddot{r}}|/c^2 \leqq \left[2gR_E/c^2\right]\left[R_E/R\right]$ which is always less than 10^{-7} .

For the m direction cosine a formula similar to (38) holds true:

$$\boxed{\frac{\lambda \Delta\varphi_m}{a} = m - \frac{\dot{y}}{c} + 0(10^{-7}) ,} \tag{39}$$

where in this case $\Delta\varphi_m$ represents the phase difference of the North-South antenna pair.

2.2 MINITRACK DATA

2.2.1 Standard Minitrack Data Format

The following is the Standard Minitrack data message that is stored on magnetic tapes by satellite and by station. The message appears on teletype output as in the following sample.

Satellite ID and date →

```
&6406401 1 690103
```

Data →

```
4350.2639114.3580.2639114.1230.2639114.00380.2639114.215.2639114.
1456.3071750.4503.3231786.1217.3121830.00334.3251831.215.3391904.
1662.3591152.4505.3581199.1222.3761243.00335.3771280.215.3731324.
1861.4081559.4504.4241597.1217.4221636.00339.4252685.215.4321729.
2061.4741972.4505.4672022.1220.4761060.00342.4812083.215.4881142.
2263.5342388.4506.5391436.1222.5301463.00347.5411519.215.5472546.
2463.5911798.4507.5912840.1227.6082886.00348.6062920.215.6102969.
2664.6502208.4507.6491255.1228.6522305.00350.6721342.215.6792385.
2863.7162633.4505.7212679.1232.7312717.00351.7372762.215.7452802.
3064.7882057.4508.7992104.1236.8062143.00354.8202186.215.8192239.
3265.8572483.4507.8672528.1238.8342574.00358.8842604.215.8961652.
3467.9482918.4507.9431950.1238.9612006.00359.9592031.215.9802078.
3668.0162345.4508.0281380.1241.0372431.00366.0482459.215.0582517.
3867.1022771.4509.1162812.1247.1242870.00365.1312904.215.1412946.
4068.1892200.4507.2011247.1249.2102297.00356.2152342.215.2252384.
4269.2831634.4510.2992684.1251.2932731.00371.3131768.215.3051830.
4469.3721086.4511.3871115.1256.3921171.00370.4031207.215.4121258.
4670.4741514.4511.4821560.1259.4921607.00376.4941661.215.5021704.
4870.5701966.4511.5801993.1261.5871055.00376.5981087.215.6031146.
5071.6691395.4513.6841450.1265.6931503.00379.6941544.215.7081598.
5271.7781848.4512.7841909.1268.7951947.00383.8001990.215.8281045.
5472.8891289.4514.8961351.1271.9131397.00385.9151439.215.9310481.
5672.9841751.4515.0030801.1273.0231826.00387.0161901.215.0611942.
5874.1140218.4516.1190238.1278.1370292.00390.1691338.215.1500394.
0074.2341658.4615.2440702.1281.2700759.00393.2860777.215.2850835.
0274.3590109.4616.3740175.1283.3990211.00397.3930246.215.4070299.
0475.4750568.4619.5000613.1287.5170665.00300.4830695.215.5330745.
0676.6200003.4619.6460072.1290.6330121.00302.6540176.215.6470231.
0879.7680522.4620.7230527.1294.7930567.00305.7900650.215.8420678.
1078.8850954.4620.8280014.1298.9310025.00308.9620126.215.9890143.
1280.1020389.4621.0980426.1200.0230478.00309.0300507.215.0990595.
```

Satellite ID Terminating Message →

```
6406401
03/1321Z JAN LWNK
```

The format for the message is as follows:

C.2

MESSAGE FORMAT

Character(s)	Contents
1, 2	second (time) of frame start
3, 4	hundreds and tens digits of east-west medium phase
5	period "." separator
6 to 8	first east-west fine phase
9	signal strength indicator (AGC)
10 to 12	first north-south fine phase
13	period "." separator
14, 15	minute of frame start
16, 17	hundreds and tens digits of east-west coarse phase
18	period "." separator
19 to 21	second east-west fine phase
22	signal strength indicator (AGC)
23 to 25	second north-south fine phase
26	period "." separator
27, 28	hour of frame start
29, 30	hundreds and tens digits of north-south medium phase
31	period "." separator
32 to 34	third east-west fine phase
35	signal strength indicator (AGC)
36 to 38	third north-south fine phase
39	period "." separator
40 to 42	day of year for frame start
43, 44	hundreds and tens digits of north-south coarse phase
45	period "." separator
46 to 48	fourth east-west fine phase

MESSAGE FORMAT (Cont'd)

Character(s)	Contents		
49	signal strength indicator (AGC)		
50 to 52	fourth north-sound fine phase		
53	period "." separator		
54	equatorial/polar antenna indicator 1 = equatorial 2 = polar		
55, 56	station number		
	03 = FTMYRS	15 = WNKFLD	01 = BPOINT
	05 = QUITOE	16 = JOBURG	13 = COLEGE
	06 = LIMAPU	19 = ALASKA	17 = MOJAVE
	08 = SNTAGO	21 = ORORAL	14 = GRDFKS
	12 = NEWFLD	23 = MADGAR	18 = WOOMER
57	period "." separator		
58 to 60	fifth east-west fine phase		
61	signal strength indicator (AGC)		
62 to 64	fifth north-south fine phase'		
65	period "." separator		

2.2.2 Extracting and Sorting Minitrack Data

The GEOS-A Minitrack data extracting program and the sorting program are special purpose programs each with a single function. The extracting program reads a Master Minitrack data storage tape and extracts only data taken on the GEOS-A satellite. This data is copied to another magnetic tape in the same format (same as given in the example in 2.2.1). This results in a data tape of GEOS-A Minitrack data grouped by station in chronological order. The FORTRAN listing of the extracting program is given in Appendix A-2.1. The requirements for the program operation are:

Function	Unit	Description
Card Input	Card Reader	Program Source Deck
Tape Input	Data Set 9 (FT09F001)	Minitrack messages for several satellites
Tape Output	Data Set 11 (FT11F001)	Minitrack messages for GEOS-A (Satellite ID 65891)
Printed Output	Data Set 6	Total number of messages Number of GEOS-A messages First 100 messages from output tape

The tape output (Data Set 11) from the extract program is the input data tape for the sort program. This program sorts messages in time-sequence. It also eliminates duplicate messages. Messages for the same station and antenna configuration are assumed to be duplicates if they commence within 30 seconds of each other. If two duplicate messages are of unequal length, then the shorter message is discarded. If they are of the same length, then the last received message is discarded.

The length of a message is determined by the number of good records it contains. The characters in each record are checked for numeric characters and periods in the appropriate places. A record with an error is discarded. The first "proper" record of a message is the calibration line. If this contains an error, the whole message is discarded. It has been found that in some messages the calibration line and subsequent records are repeated within the same message. For this reason, every record is compared with the calibration line. If a record is found to be identical to the calibration line, then this is regarded as the first "proper" record of the message. All previous "proper" records are ignored.

Messages containing fewer than 5 records are discarded. The output tape from the sort program is in the same format as the input tape.

The FORTRAN listing of the sort program is given in Appendix A-2.2.

The requirements for running the sort program are as follows.

Function	Unit	Description
Card Input	Card Reader	Source Program
Data Cards	Data Set 5	17 Cards Required (See Appendix A-2.2)

Cards 1-16 (One card for each Station)

Card Column	Word	Definition	Format
1-5	KSTA	Station ID	I5
6-10		Blank	
11-16	STATIO	Station Name	A6

Card 17

Card Column	Word	Definition	Format
1-8	INTAPE	Input tape No.	A8
9-13	INFILE	File No.	I5
14-21	NDTAPE	Output tape No.	A8
22-26	NDFILE	File No.	I5

Function	Unit	Description
Tape Input	Data Set 9 (FT09F001)	Input tape containing Minitrack messages (Output tape from Extract program)
Tape Output	Data Set 10 (FT10F001)	Output tape containing sorted Minitrack messages
Printer Output	Data Set 6	A list of message numbers (as sequenced on input tape) of rejected messages together with reasons for rejection

A list of output information giving the following. (The actual data message is not printed out):

- Sequence number on output tape
- Sequence number on input tape
- Station ID (KSTA)
- Station Name (STATIO)
- Day Number (1 to 366)
- Month
- Day of Month (not correct for leap year)
- Seconds of day
- $\left.\begin{array}{l}\text{Hours}\\ \text{Minutes}\\ \text{Seconds}\end{array}\right\}$ (of day)
- Number of lines in message

2.2.3 The Minitrack Preprocessor

The purpose of the Minitrack preprocessor is to make known corrections to the Minitrack data and output a magnetic tape containing corrected data in a format acceptable to the NAP-II program. Basically, the preprocessor reads the raw phase data message, computes phase differences, makes corrections and then converts the corrected phase differences to direction cosines. This process is explained in detail in references 1 and 2.

The Minitrack preprocessor used was based on that developed by the Goddard Space Flight Center Network Computation Section (references 1 and 2). The program was extensively modified by W. M. Rice (GSFC, Code 551). A few minor modifications was then made by DBA Systems, Inc. under this contract. A complete listing of the preprocessor program is given in Appendix A-2.3. Sample JCL for running the program is also given.

> Changes Made by W. M. Rice. A Minitrack message contains, in general, 155 measurements of phase differences between two antenna pairs. The preprocessor converts these to 155 pairs of direction cosines. A quadratic fit is then made to each set of 155 direction cosines. The resultant output is a single pair of direction cosines (the fitted mid-point values). The program is made more complicated by the fact that the ambiguity of the direction cosines must be resolved using additional information in the Minitrack message (Reference 2). However, this is only done to the fitted midpoint values. In the Rice version of the program the ambiguity is resolved for all direction cosines. The output consists of all direction cosines and is in a format that can be used as an input to the NAP program. The time associated with each measurement is given at the instant that the signal being analyzed arrived at the relevant antenna pair.

Changes Made by DBA Systems, Inc. To facilitate the data preparation for the NAP program, the Minitrack preprocessor was modified to punch out the three control cards (Categories 201, 202, and 999) required for each data message. (For a description of the required control cards, see reference 4).

Although nominally there is a single time associated with each pair of direction cosines, this is not so in practice. Differences arise due to different filter delays associated with each direction cosine and also due to different counter delays (reference 3). As the volume of printed output from the NAP program is proportional to the total number of different time points, the output volume could be reduced by one half, if each pair of direction cosines were adjusted to the same time point. This was done. The adjusted time point was chosen as the average time of the two measurements. For a typical NAP run, this reduced the output volume by half.

A modification was also made to output on tape in a NAP input format, the fitted midpoint values of the original program. The idea behind this was that the fitted midpoint values could be used initially as an input to NAP, the "all data points" input being reserved for the final iterations through NAP. This in theory should have reduced the total computer time required. In practice, however, it was found that the "all data points" input when processed through NAP, provided the user with an extremely useful criterion for determining the quality of the data in a Minitrack message -- the standard deviation of the error of all measurements within a message. For this reason, little use was made of the "fitted midpoint" input.

To help in the analysis of the results of the NAP program, a modification was made to print out a summary of the results of the preprocessing. This summary contained the following information for each station: For each data message the midpoint time, the three midpoint direction cosines and their rates of change, plus the prepass calibration constants.

Finally, an error in the way the 2 Hz filter delay was being applied was corrected.

> A Program Error. Messages spanning midnight are not handled correctly by the program. A Minitrack message typically spaces a time interval of one minute or one half of a minute, so that this occurs very infrequently, but the program error should be corrected.

The Minitrack preprocessor exists as a modified version of the original program developed by GSFC. Some of the information used in the original program is not used in its present form. The input cards to the program, however, are the same as with the original program. Consequently, a large number of cards containing little information are required.

The following information is required for operation of the program.

Function	Unit	Description
Card Input	Data Set 5	Station Calibration Constants
		Card 1 Blank
		Card 2 Station Constants

Card Column	Word	Definition	Format
1-6	STATIO	Any 6 alphanumeric characters	A6
7		Blank	

Function	Unit	Description			
		Card Column	Word	Definition	Format
		8, 9	KSTA	Station number	I2
		10-13	KFA	Time adjustment EW-channel	I4
		14-17	KFB	Time adjustment NS-channel	I4
		18-20		Blank	
		21-24	EWM	Phase bias medium antenna, EW-channel	F4.3
		25-27	CLEWM	Cable length in-equality, EW-medium channel	F3.3
		28-31	EWC	Phase bias coarse antenna, EW-channel	F4.3
		32-34	CLEWC	Cable length in-equality, EW-coarse channel	F3.3
		35		Blank	
		36-39	EWFEQ	Phase bias equatorial fine antenna, EW-channel	F4.3
		40-43	EWFPO	Phase bias polar fine antenna, EW-channel	F4.3
		44-46		Blank	
		47-50	NSM	Phase bias medium antenna, NS-channel	F4.3
		51-53	CLNSM	Cable length in-equality NS-medium channel	F3.3
		54-57	NSC	Phase bias coarse antenna, NS-channel	F4.3
		58-60	CLNSC	Cable length in-equality NS-coarse channel	F3.3
		61		Blank	

Function	Unit	Description			
		Card Column	Word	Definition	Format
		62-65	NSFEQ	Phase bias equatorial fine antenna, NS-channel	F4.3
		66-69	NSFPO	Phase bias polar fine antenna, NS-channel	F4.3
		70-74		Blank	
		75-79	DATE	Date of calibration	I6
		Cards 3 thru 10		Blank	
		Cards 11-135		15 sets of cards 2 thru 10 are required. One set for each of the Minitrack stations	

Card 136 Card Column	Word	Definition	Format
1-5	KSAID	Satellite ID	I5
6-24		Blank	
25-32	FREQ	Satellite transmitter frequency (MHZ)	F8.3
Card 137		Blank card terminates data set.	

Function	Unit	Description
Card Input	Data Set 8	Data Selection Control Cards

Card 1 — 80 alphanumeric characters used to write a message in the pre-processed printout

Card 2 — (more than one card can be used)

Card Column	Word	Definition	Format
1-10	ILOW, IHIGH	Program will process messages between ILOW and IHIGH	2I5

Card Column	Word	Definition	Format
11-15	IYEAR	2-digit integer for year of measurements e.g. 1966 = 66	I5
16-20	NEWARC	Arc number assigned to measurements. If several ILOW, IHIGH cards define some arc, NEWARC should be blank on all but first card.	I5
21-25	NAPEND	Normally left blank -1 = more data tape to follow 0 = normal termination 1 = no terminal record written (for tape addition) According to Ref. 4, the input tape to NAP should terminate with a negative number in the last record. However, with current versions of NAP this is no longer necessary. If NAPEND is left blank, this terminal record is written on the tape. If on the last "ILOW, IHIGH" card NAPEND is set equal to a positive non-zero integer, then the terminal record is not written. (The reason for this option is to facilitate combination of two or more data tapes into a single tape). If the messages being processed are contained on more than one tape, then NAPEND should be set to a negative integer on the last "ILOW,IHIGH" card referring	

Card Column	Word	Definition	Format
		to a tape that is to be followed by another tape. e.g., if data messages to be processed are contained on three tapes such that FT09F001 defines tape 1, FT09F002 defines tape 2, FT09F003 defines tape 3, then the last "ILOW,IHIGH" card of tape 1 should have a negative NAPEND, the last "ILOW,IHIGH" card of tape 2 should have a negative NAPEND, the last "ILOW,IHIGH" card of tape 3 should have either a blank or a positive NAPEND.	
Card 3, 4		Blank Each "ILOW,IHIGH" card should be followed by two blank cards. The program actually has a restart capability. The blank cards must be inserted if the user does not wish to avail himself of the re-start facility. For use of the restart facility the user is referred to the program listing, Appendix 2.3	

Function	Unit	Description
Tape Input	Data Set 9	Data tape(s) containing Minitrack messages to be processed
Tape Output	Data Set 12	Preprocessed output of "all data points" in a NAP-II input format
Tape Output	Data Set 19	Preprocessed output of "fitted midpoint" in NAP-II input format

Function	Unit	Description
Punched Card Output	Data Set 7	NAP-II control cards for each data message. These are the start-stop times for that station data set (NAP-II cards group 201, 202, 999)
Printed Output	Data Set 6	Raw Minitrack messages with each message followed by the computed direction cosines for each data point. (Direction cosines are also given in terms of Minitrack counts.)
	Data Set 11	Printout of intermediate results obtained in preprocessing
	Data Set 13	Input station constants (See data set 5) List of all messages that have not been preprocessed and reasons why. Pass summary information (number of messages per station, time of last message for each station) Information required for restart.
	Data Set 14	Summary of preprocessed results arranged in time sequence (station, time, direction cosines, direction cosine rates)
	Data Sets 20 thru 51	32 data sets -- 1 per station. Although there are only 16 stations the program treats a station operating in the equatorial mode as one station with station number KSTA (See data set 5) and the same station operating in the polar mode as a different station with station numbers KSTA+100. Each data set contains preprocessed results (for each station) arranged in time sequence (time, direction cosines, direction cosine rates, prepass calibration constants.)

Suggested Running Time

For the 360/91	CPU time	11 minutes per 1000 messages
For the 360/95	CPU time	14 minutes per 1000 messages

2.3 Optical Data

2.3.1 Standard Optical Data Format

The optical data used in this study was obtained from the NASA Space Sciences Data Center (NSSDC) at Goddard Space Flight Center. The data was on magnetic tape in the "GEOS format." A complete description of the data format is available from the NSSDC. The "GEOS format" for optical data is basically 80 column card images stored on magnetic tape. Each card image contains the following.

Card Column	Description
1-6	Satellite identification
7	Type of coordinates (RA and DEC)
8	Observation identifier
9-11	Timing standard deviation
12-13	Time identifier
14-18	Station number
19-34	GMT of observation
35-53	Observation data
54-59	Date of plate reduction
60-71	Code information as to processing
72-80	Description of random error

A separate data tape for each tracking network is normally obtained from the NSSDC. For example, in this study optical data from the STADAN network and the SAO network was to be used. Two data tapes were obtained, one for STADAN and one for SAO. In order to simplify the optical preprocessing, it was necessary to merge the two tapes. A special program was written to this task. The program listing is given in Appendix A-2.4. The program merges two time sequenced optical data tapes into a single tape which will be the input data tape for the optical pre-processor. Data for each month is written on a separate file numbered sequentially on the same tape.

The following is required to execute the merge program:

Function	Unit	Description
Card Input	Data Set 5	Tape and file number of output tape used for printout.

Card Column	Word	Definition	Format
1-8	TAPE	8-character tape number	A8
9-10		Blank	
11-15	IFILE	File number of first month of the combined tape	I5

Function	Unit	Description
Tape Input	Data Set 1	Data tape with STADAN stations.
Tape Input	Data Set 2	Data tape with SAO station.

NOTE: The user should examine the printed output of the input data tapes to determine if there are dummy or non-essential data records on the tapes.

Function	Unit	Description
Tape Output	Data Set 3	Merged data output (Input tape to Optical Preprocessor). JCL cards required for each month there is data. FT03F001 should have tape and file no. corresponding to card input, data set 5. The reason for writing a separate file for each month is to minimize tape search required in the optical preprocessor.
Printed Output	Data Set 6	Printed output of the merged output data tape (data set 3). Each page has a heading giving the year and month of the data and the tape and file number on which the data may be found on tape.

2.3.2 Optical Data Preprocessing

The program assumes that the data on the input tape (data set 1) is arranged in time sequence. The program resequences the data such that the data for each station/orbit combination is sequential. An orbit is defined as consisting of TORBT seconds (45 minutes). Furthermore, each "station/orbit" is subdivided into passes (photographic plates) where each pass may be TPASS seconds (45 seconds) long. The data itself is converted to radians and output on data set 2 in the NAP input format. NAP control cards are punched (data set 7) for each pass.

For SAO stations (station ID 29XXX), the input data is given in the 1950.0 coordinate system. This is converted to the "true of date" coordinate system. The precession and nutation formulae used are based on those given in the Explanatory Supplement to the Astronomical Ephemeris and the American Ephemeris and Nautical Almanac (Her Majesty's Stationary Office, London, 1961). Also, the time of the SAO stations is given as Atomic time. This is converted to UTC using information provided by the user (data set 4). A further complication arises in the case of the SAO stations because, for those stations, the time associated with the data is the time of observation rather than the time that the signal left the satellite. An "r/c" correction should thus be applied to the timing of the orbit determination program. Unfortunately, NAP-II does not possess this capability. A temporary fix has, therefore, been made (subroutine SAOCOR) utilizing the fact that, nominally, the GEOS satellite always flashes on the even second. Since this is not exactly true, it is recommended that the user compare the output time for the SAO stations with those of other stations and make the appropriate adjustments on NAP "704" cards (see reference 4).

The following cards are required for running the optical preprocessor.

Function	Unit	Description
Card Input	Data Set 5	Station and data set control cards.

Function	Unit	Description

Station Cards: Up to 30 observing stations may be used. One card for each station.

Card Column	Word	Definition	Format
1-5	KSTA	Station No. given on NAP 201/202 cards.	I5
6-10		Blank	
11-18	STNAM	8-alphanumeric character station name.	A8
19-20		Blank	
21-25	ISTA	Station ID	I5
26-30	IEND	Blank except for last "station" card where a negative integer is used to indicate last station card.	I5

Data Set Cards: One card for each arc of data.

Card Column	Word	Definition	Format
7-5	NARC	Arc number required for NAP control cards.	I5
6-10		Blank	
11-16	IYMDB	Year, month and day of beginning of arc (e.g. 660312).	I6
17-19	IHB	Hour of beginning of arc.	I3
20-22	IMB	Minute of beginning of arc.	I3
23-32	SECB	Seconds of beginning of arc.	D10.0
33-38	IYMDE	Year, month and day of end of arc.	I6

Function	Unit	Description

Card Column	Word	Definition	Format
39-41	IHE	Hour of end of arc.	I3
42-44	IME	Minutes of end of arc.	I3
45-54	SECE	Seconds of end of arc.	D10.0

Card Input — Data Set 4 — Correction from A.1 to UTC time. Required only if SAO stations are to be processed. Any number of these cards may be used and the program will use a linear interpolation to any required time. At least two cards required.

Card Column	Word	Definition	Format
1-2	IYR	Year	I2
3-4	IMO	Month	I2
5-6	IDA	Day	I2
7-10		Blank	
11-20	COR	A.1 minus UTC on year, month, day (IYR,IMO,IDA)	F10.5

Function	Unit	Description
Tape Input	Data Set 1	Data tapes to be processed (output from Merge or GEOS formatted data tape). If more than one data tape is used, then JCL will be modified to accommodate (e.g. FT01F001 for first tape, FT01F002 for second tape, etc.).
Tape Output	Data Set 2	Preprocessed data tape to be input to NAP-II.
Printed Output	Data Set 6	Print of preprocessed data.
	Data Set 3	Printed output of input (data sets 4 and 5).
	Data Set 12	Summary of data.
Punched Output	Data Set 7	NAP Control Cards for timing/station information, NAP Cards 201, 202, 999 (see Reference 4).

2.4 REDUCTION OF MINITRACK DATA

2.4.1 PRENAP Program

The processing of 20 days of Minitrack data requires about 5000 input cards. In order to reduce the number of cards that must be handled for a NAP-II run, the NAP-II control cards are written on magnetic tape. This "control card" tape is then updated and communicated to the NAP-II program via the PRENAP program. In this mode of operation, execution of the PRENAP program becomes the first step of a complete NAP-II run.

The following is a listing of the program to copy the NAP-II control cards onto magnetic tape.

```
Job Card
// EXEC FORTRANG
//SOURCE.SYSIN DD*
      IMPLICIT REAL * 8 (A-H,0-7)
      DIMENSION KAT(2),KEY(10),DATA(2)
 100  FORMAT (13,12,A8, 1013,D22.8,D15.8)
      DO 300 I=1, 10000
      READ (5, 100,END=400) KAT,XLABEL,KEY,DATA
      WRITE (9) KAT,XLABEL,KEY,DATA,I
 300  CONTINUE
 400  ENDFILE 9
      REWIND 9
 200  FORMAT (1X,3I8,A8, 10I3,2D19.8)
      DO 330 J=1, 10000
      READ(9,END=500) KAT,XLABEL,KEY,DATA,I
      WRITE(6,200) I,KAT,XLABEL,KEY,DATA
 330  CONTINUE
 500  STOP
      END
```

```
/*
// EXEC LOADER,PARM='MAP,CALL,SIZE=100K',REGION.GO=110K
//GO.FT09=001 DD UNIT=9TRACK,LABEL=(4,BLP),DSN=MRGDAP,DISP=(NEW,KEEP),
// DCB=(RECFM=VBS,LRECL=80,BLKSIZE=3204) VOL=SER=34517C
//GO.DATA5 DD*
```

Once the NAP-II cards are written on tape, the function of the PRENAP program is to update the "cards" on tape and to convert the binary "cards" tape to a format acceptable to the NAP-II program. The PRENAP program is listed in Appendix A-2.6. At the end of the program listing is an example of the JCL cards needed and a sample set of update cards.

The program is normally loaded as a binary object deck (Appendix A-2.6 JCL cards). The function of the peripheral equipment needed to execute the PRENAP program are:

Function	Unit	Description
Input Tape	Data Set 9	NAP-II control cards on magnetic tape. (Output from program given above).
Disk File	Data Set 10	Used as an intermediate scratch tape.
Disk File	Data Set 12	Used to pass "updated" NAP-II control cards to NAP-II program for execution.
Card Input	Data Set 5	Updater cards for NAP-II control cards. If no update is required, data set 5 should be left empty. There would probably be no update the first time NAP-II is run for a particular job. The program prints out instructions on how to update "cards".

2.4.2 Use of NAP-II Program

The NAP-II program is a very flexible analysis tool providing the user with considerable freedom for designing and executing data reductions. Consequently, the user must have a good understanding of the program uses to attain efficient utilization. This can only be done through use of the program and its various options. The functions of the program are discussed in reference 4.

The type of information required to execute NAP-II is given as follows:

Type of Information	Categories
General and planetary information	101, 102, 103, 151
Comments	150
Station survey	301, 302, 303
Totally stable parameters	601, 602
Measurement definitions	701, 702, 703, 704 (Sets 0 only)
General arc information	104, 205, 206
Arc stable parameters	601, 602
General pass information	201, 202, 203, 204
Pass stable parameters	601, 602
Pass comments	152
Overrides for measurement definitions for a pass	701, 702, 703, 704 (Sets 1 only)
End of pass, arc, or all control data	999

The station numbers (category 300) are preassigned by the Minitrack and the optical data preprocessors, depending on the order of the Minitrack Station Calibration Cards and the Optical Stations Control Cards. This controls both the station ID numbers and the user assigned station numbers. These two programs also provide the timing information cards (category 200 and 999) as part of their output. The other cards must be set up by the user. The information required by the various groups of cards is given in reference 4.

Once a "master" set of cards has been set-up, subsequent runs can be made with new data sets by changing the following quantities,

- State vectors - Category 601, 205
- Greenwich Hour Angle - Category 206

- Lunar and Solar Coordinates - Category 104
- Timing information - Output from data preprocessors.

These cards can be changed via the PRENAP program.

To run the NAP program, it is essential to have a reasonably good estimate of the state vector at the initial epoch. Before the user tries to solve for any parameters, he should make an initial run, which will allow the POSTNAP program to edit out obviously bad data and correct the lobe assignments made by the Minitrack preprocessor. Once this has been done, the user can go ahead and solve for any parameters he wishes. As the user improves his estimate of the initial state vector, he may wish to modify his definition as to what constitutes bad data.

The NAP-II program outputs the following information on disk or tape:

measurement discrepancies (FT37F001)

current estimates of error model parameters (FT37F002)

data tape with the mid-data point for each pass (FT37F003) .

For multi-arc data processing, it is recommended that the user switch from using the data tape output from the Minitrack preprocessor to the "single data point per pass" data tape. When using the single data point prepass, the Category 704 cards have to be deleted, because the biases have already been applied. The run time advantage of using one data point instead of all data points (which may be as many as 155) is considerable. The loss of accuracy is negligible.

The recommended procedure for reducing Minitrack data is summarized in Section 2.6.

There were changes made to the NAP-II program to effect the procedure for reducing Minitrack data. The need for some of these changes was discussed in the preface of this report. Without going into great detail, the changes to the particular subroutines are as follows:

- Changes to Subroutine ENEXPS

  This subroutine computes two sets of series: (1) the power series expansions of n-heavenly bodies, and (2) the power series expansion defining whether or not the satellite is exposed to solar radiation. The subroutine was completely reprogrammed, making the new version seven times faster than the old. One of the factors contributing to the increased computational speed of the new program is that it computes the n(n-1)/2 different distances between n bodies, and no more. The old program somehow managed to compute more distances than this so that some distances had to be identical.

  The old program had a programming error in the computation of the "shadow" series so that the two programs do not agree as far as shadow series expansion is concerned. The FORTRAN listing of the subroutine is given in Appendix A-3.1.

- Changes to Subroutine EXPAND

  Subroutine EXPAND computes two sets of series: (1) the power series expansion of the acceleration due to the full gravity field, expressed as a sum of spherical harmonics, of the main attacking body, and (2) the power series expansion of the acceleration due to atmospheric drag. The part dealing with the gravity field was completely rewritten. The mathematical analysis for this is given in reference 5. The FORTRAN listing of the subroutine is given in Appendix A-3.2.

- Changes to Subroutine VARIEQ

  Subroutine VARIEQ computes the two sets of series: (1) the power series expansion of the variational matrix due to the full gravity field of the main attracting body, and (2) the power series expansion of the variational matrix due to atmospheric drag. The FORTRAN listing of the subroutine is given in Appendix A-3.3.

- Changes to Subroutine FINALP

  A small change was made in this subroutine to output on disk (data set 10) the computed values for the error model terms. Note that the call to this routine precedes the final call to subroutine RESID so that the quantities output by this routine on data set 10 precede those output by RESID. The FORTRAN listing is given in Appendix A-3.4.

- Special Subroutine PRTIAL

  The standard subroutine PRTIAL computes the discrepancies between observed and predicted measurements and also the partial derivatives of the predicted measurements with respect to those error model terms that affect the predicted measurements. The computed values are then stored on a disk file (ISFILE). Later in the program subroutine SOLVER forms the normal equations from this disk file and solves the equations to give the computed values for the error model terms. The quantities on the disk file are also used by subroutine RESID which prints the measurement residuals. (The residuals are updated discrepancies between observed and predicted measurements, where the predicted measurements reflect the latest values (from SOLVER) of computed error model terms. The quantities involved in this computation are the discrepancies and partial derivatives from subroutine PRTIAL, which are stored on disk file, and the difference between the computed values for the error model terms and the values used by subroutine PRTIAL for the predicted measurements).

  Obviously, the more data points that are stored on ISFILE the longer it will be. For a one-week long arc of Minitrack data, ISFILE is about 3,000,000 bytes long (20 cylinders of disk space).

To print the final residuals would require a further 9,000,000 bytes (60 cylinders of disk space). Together with the remaining disk requirements of the NAP program this puts a severe strain on the total 360 system.

To make it practicable to use the NAP program for multi-arc cases, several changes were made to subroutine PRTIAL.

> Changes to Subroutine PRTIAL. For each measurement all discrepancies for a pass (Minitrack message) are averaged to give the mean measurement discrepancy. The standard deviations are also computed and any discrepancies differing from the mean by more than three standard deviations are rejected. (The process is iterative, since as soon as a discrepancy is rejected a new mean and standard must be computed). The mean discrepancies are then output on ISFILE together with the appropriate partial derivatives. The time associated with the mean discrepancies is also computed. These measurements are written on the same tape on disk file as the discrepancies (FT37F001). They are also written on disk (FT35F001) in a NAP observation tape format. The standard deviations and the number of measurement points (not rejected) are written on disk (FT36F001). The FORTRAN listing of the subroutine is given in Appendix A-3.6.

- Changes to Subroutine RESID
  - There are two subroutines RESID that can be used in NAP-II. These are identified as STANDARD Subroutine RESID and SPECIAL Subroutine RESID. The Special subroutine was written to make multi-arc runs practical in terms of run time.

Standard Subroutine RESID. Three sets of changes were made to this subroutine. The new subroutine will work only if there are two or fewer kinds of measurements per station-pass. The first change involved writing the printed (as on data set 6) pass summary on disk (data set 10) for use by some postNAP program, and also on data set 34, which is a printed pass summary. The second change involved plotting the measurement residuals in addition to giving the numerical values. The third change involved the computation of the correlation coefficient between the errors of the two measurements. The reason for computing the correlation coefficient was that when the plotting facility was added to this subroutine, it looked as if there was some correlation between the two measurement errors. However, in most cases the correlation coefficient is very small. The FORTRAN listing of the subroutine is given in Appendix A-3.5.1.

Special Subroutine RESID. The computed values of the error model terms are copied from disk (FT10F001) and written on tape (FT37F002). The measurement residuals are computed as before. However, since there now is only one measurement time point per pass, the standard deviation cannot be computed. Instead, the standard deviation computed in subroutine PRTIAL is copied from disk (FT36F001) as is the number of measurement points. This information is written on the same tape as the new values for the error model terms. Finally, the new observation tape is copied from disk (FT35F001) to tape (FT37F003). The FORTRAN listing is given in Appendix A-3.5.2.

- Changes to Subroutine MESOLD

  Terms were added to Subroutine MESOLD to reflect the complete Minitrack geometrical error model. The following should be added to the NAP-II User's Guide (Reference 4) to be consistent with these changes (Appendix IV, Page IV-2).

MEASUREMENT CODE \ ERROR MODEL TERM NO	12	15	16
6		RE	RM
7	TM	RP	RL

  where measurement code 6 corresponds to direction cosine alpha (or ℓ), and measurement code 7 corresponds to direction cosine beta (or m), and $n = \sqrt{1-(\ell^2+m^2)}$

  The 12th error model term (TM) is not part of the geometrical error model but corresponds to a timing bias.

  RE and RP are rotation terms about the local vertical of the equatorial and polar m measurements, respectively. RL and RM are rotations or tilt about the ℓ or m measurements.

2.4.3 The NAP-II Minitrack Post-Processing Program

The post-processing program summarizes the results of a NAP-II Minitrack run and prepares the NAP-II "input cards" for the next iteration. Any lobe assignment errors of the Minitrack preprocessor can also be corrected through the NAP-II "input cards".

In order to execute the post-processing program, the NAP-II program must be a version employing the special subroutine PRTIAL and RESID given in Appendix A-3.6 and 3.5.2, respectively.

The program assumes that the station numbers assigned by NAP are odd for equatorial stations and even for polar stations and that measurement numbers are odd for "L" measurements and even for "M" measurements. (See Appendix A-4.)

The NAP-II control cards must be in a particular order required by the POSTNAP program. In general, error model terms should be setup in order of totally stable, arc stable and pass stable. The error model parameter numbers (Category 601, Key 7), must be in ascending and consecutive order. Cards must not be repeated.

This is not a restriction on the NAP-II program but a restriction imposed by POSTNAP.

Function	Unit	Description
Card Input	Data Set 5	Only one card required.

Card Column	Word	Definition
1-5	IOPT	Determines which NAP "701" (continuation 1) cards are to be output. If measurements were previously edited out of the solution and no value for the pass standard deviation is available, the measurements are edited out again. IOPT can assume values of 0, -1 and +1. In all three cases the above applies. In all cases key 7 of 701 card is set to "1.
		= 0, measurements are edited out if mean pass measurement error exceeds MAXERR or if pass standard deviation exceeds SDMAX. If measurements are edited out, DATA 2 of 701 card is set to 1.D15.
		= -1, measurements are edited out if mean pass measurement exceeds MAXERR or if no pass standard deviation is available or if pass standard deviation exceeds SDMAX. If measurements are edited, DATA 2 is set to 1.D15.
		= +1, same as -1 case except if measurements are not edited out, they are assigned "a priori" sigma of 10 times pass standard deviation. If measurements are edited out, DATA 2 is set to 1.D7 times the pass standard deviation.

Function	Unit			Description
		Card Column	Word	Definition
		6-10	MAXERR	Maximum mean measurement error in Minitrack counts.
		11-20	SDMAX	Maximum pass standard deviation (unscaled).
		21-30	FREQCY	Satellite transmitter frequency in MHz.

Note on the availability of a Pass Standard Deviation.

A pass standard deviation can only be computed if there is more than one measurement per pass. If IOPT = 1, the pass standard deviation is saved on a "701" card. (If key 7 =0 , it is equal to DATA 2 times 1.D-1; if Key 7 = 1, it is equal to DATA 2 times 1.D-7). Use is made of this facility when the data processing switches from using all measurements for a pass to the single new measurement computed in special subroutine PRTIAL.

Note on data editing.

A bad pass is defined as one whose mean error exceeds MAXERR or whose standard deviation exceeds SDMAX. Bad passes are edited out of the solution in all cases. If no standard deviation is available, then in the case of Iopt = -1 the pass will be edited out of the solution. (It is recommended that Iopt = -1 only be used if there are more than one data point per pass. The non-availability of a pass standard deviation may in that case be the result of the pass not meeting the elevation requirements. NAP in that case would output a zero mean pass error).

Iopt = 1 should be used if the user wishes to weight the measurements of a pass in inverse proportion to their standard deviation, in other words, a noisy pass would be given less weight than one that is less noisy.

Function	Unit	Description
Disk Input (on tape)	Data Set 27	NAP input cards used in making NAP run. This would normally be on a disk file created by the PRENAP card updating program.
Disk input (on tape)	Data Set 10, File 1 (FT10F001)	Output from NAP data set 37 (FT37F002). The computed values of error model parameters.
Disk input (on tape)	Data Set 10, File 2 (FT10F002)	Output from NAP data set 37 (FT37F001). Current measurement discrepancies.
Disk output (on tape)	Data Set 10, File 3 (FT10F003)	Output of sequenced NAP "input cards" for use by the PRENAP card updater program on the next iteration.
Printed output	Data Set 6	Prints the following • Some instructions on the use of the program. • Printout of input cards (data set 5). • Printout of NAP "output cards" (FT10F003) for the next iteration. The first column of this list are the sequence numbers.
Printed output	Data Set 20	The following is printed out for each station/pass. • Time of midpoint of pass. • Mean measurement error (This is the mean discrepancy computed in special subroutine PRTIAL and equals the mean residual error computed in subroutine RESID when the solution converges). • Standard deviation of the measurements for a pass. For only one measurement the standard deviation is equal to zero. • For each measurement, the time relative to mid-point and the error relative to mean pass error. This error is also plotted.
Printed Output	Data Set 31	Printout of the bias values used in the Minitrack preprocessor (built into the program) and the new bias. New biases are obtained by converting the bias obtained by NAP to Minitrack counts and adding the result to the preprocessor bias.

Function	Unit	Description
Printed output	Data Set 31-60	A summary for each station is written on a different data set. Each pass contains • Time of midpoint measurement • Arc number • Pass number • Mean error in Minitrack counts • Whether ( ) or not (*) the measurement met elevation requirement given on NAP "702" card • Whether ( ) or not (*) measurement contributed to present solution • Measurement standard deviation in Minitrack counts • Bias correction made to the measurement for the next iteration expressed in Minitrack lobes (in 1000 Minitrack counts). The errors being corrected are due to wrong lobe assignments in the Minitrack preprocessor and are corrected via NAP "704" cards • Number of measurement points per pass on the data tape • Number of measurement points deleted by special subroutine PRTIAL • The mean error rate • The reduction in standard deviation, expressed as a percentage, that would result if the standard deviation were computed about the line defined by the mean error and mean error rate and not the mean error alone • The computed measurement. The predicted (NAP) measurement plus the mean pass measurement error, (computed in special subroutine PRTIAL). • Whether ( ) or not (*) the measurement will contribute to the solution of the next iteration.

2.5 RESULTS OF MINITRACK DATA REDUCTIONS USING NAP-II

2.5.1 Estimation of Calibration Parameters for the Fort Myers Minitrack Station Using the GEOS-I Satellite and Optically Determined Reference Arcs

A number of GEOS-A arcs determined from optical observations were used as reference arcs. These arcs had previously been obtained by DBA Systems, Inc. for work on another contract. Since optical observations are considerably more accurate than Minitrack, the optical reference arcs could be considered perfect. The NAP program was then used to recover measurement biases and measurement scale factor errors. The Minitrack measurements for each pass were weighted in inverse proportion to the variance of the measurement error for each pass (Minitrack message).

The results are shown in Tables 2 and 4. Tables 1 and 3 give the actual measurements. Table 5 is a comparison of calibration results. Table 6 tabulates some observed frequencies. It shows that the fluctuation in the GEOS-A transmitter frequency is very small. Table 7 shows the effect of including the Doppler effect and wave propagation time delay in the computations. The total effect can be seen to be of the order of 1 Minitrack count.

TABLE 1

POLAR ARRAY. Observed values of direction cosines, direction cosines rates and prepass internal bias (KS2).

ARC Number	ORBIT Number	DATE (1966) MTH	DATE (1966) DAY	GMT HRS.	GMT MIN	Direction Cosines l	Direction Cosines m	Direction Cosines n	Direction Cosine Rates $\dot{l}$	Direction Cosine Rates $\dot{m}$	Direction Cosine Rates $\dot{n}$	KS2 (l) - 067	KS2 (m) - 560
1	700	1	4	6	23	.84	.00	.55	.0009	-.0030	-.0014	0	0
2	712	1	5	6	27	.63	.01	.73	.0017	-.0037	-.0015	0	2
3	784	1	11	6	50	-.87	.00	.49	.0004	-.0031	.0007	2	14
5	831	1	15	5	4	-.21	.00	.98	.0026	-.0047	.0006	2	0
7	843	1	16	5	7	-.52	.00	.85	.0017	-.0043	.0010	2	4
9	855	1	17	5	11	-.72	.00	.69	.0009	-.0037	.0010	0	1
10	866	1	18	3	14	.86	.07	.51	.0010	-.0027	-.0013	2	2
11	867	1	18	5	16	-.84	.00	.54	.0005	-.0032	.0008	3	2
12	914	1	22	3	30	-.10	.00	.99	.0027	-.0046	.0003	2	8
15	938	1	24	3	37	-.67	.00	.74	.0011	-.0038	.0010	2	0
16	949	1	25	1	40	.90	.00	.44	.0006	-.0024	-.0013	3	0
17	961	1	26	1	44	.80	.01	.61	.0011	-.0029	-.0014	1	-1
18	1080	2	5	0	20	.11	.00	.99	.0027	-.0043	-.0003	2	10
19	1092	2	6	0	25	-.25	.00	.97	.0023	-.0043	.0006	4	3
20	1408	3	4	9	19	-.43	.00	.90	.0012	.0022	.0006	6	-41
21	1479	3	10	7	40	-.13	.00	.99	.0014	.0024	.0002	4	-51
22	1680	3	27	2	40	.76	-.12	.64	.0005	.0021	-.0003	16	14
23	1681	3	27	4	41	-.63	.00	.77	.0010	.0020	.0008	16	14
24	1740	4	1	3	0	-.15	.00	.99	.0015	.0025	.0002	18	15
25	1752	4	2	3	3	-.38	.02	.93	.0014	.0023	.0005	18	15

TABLE 2

POLAR ARRAY. Measurement Errors.

ARC No.	Standard Deviation Of Measurement Error About Mean For Pass σl	σm	Mean Measurement Error For Pass Δl	Δm	MINITRACK COUNTS Δl	Δm
1	.000072	.000147	-.000001	-.000135	0	-8
2	.000036	.000034	-.000038	-.000097	-2	-6
3	.000749	.000660	-.000164	-.000034	-9	-2
5	.000068	.000072	.000027	.000085	2	5
7	.000040	.000063	.000009	.000033	1	2
9	.000044	.000056	.000116	-.000005	7	0
10	.000239	.000141	-.000042	-.000023	-2	-1
11	.000261	.000353	.000033	-.000057	2	-3
12	.000044	.000032	.000155	-.000068	9	-4
15	.000071	.000081	.000048	.000125	3	7
16	.000133	.000173	.000112	.000029	6	2
17	.000042	.000048	.000011	.000005	1	0
18	.000035	.000034	.000003	.000098	0	6
19	.000044	.000057	-.000057	.000136	-3	8
20	*	.000038	*	.000036	*	2
21	.000052	.000048	.000087	.000047	5	3
22	.000085	.000092	.000130	.000118	7	7
23	.000029	.000032	.000003	.000018	0	1
24	.000039	.000034	-.000046	-.000005	-3	0
25	.000021	.000028	-.000062	-.000066	-4	-4

EW - BIAS: 137 MINICTS - .000021 = 136 MINICTS

NS - BIAS: 431 MINICTS +.000071 = 435 MINICTS

EW - SCALE FACTOR: -.00022

(Base Line is Shorter Than Assumed.)

TABLE 3

EQUATORIAL ARRAY. Observed values of direction cosines, direction cosine rates and prepass internal bias (KS2).

ARC Number	ORBIT No.	DATE(1966) MTH	DAY	GMT HRS.	MIN.	Direction Cosines l	m	n	Direction Cosine Rates $\dot{l}$	$\dot{m}$	$\dot{n}$	KS2 (l) - 067	KS2 (m) - 560
1	700	1	4	6	18	.00	.88	.48	.0020	-.0007	.0013	-1	0
2	712	1	5	6	23	.00	.78	.62	.0022	-.0013	.0017	-1	2
4	819	1	14	4	59	.00	.26	.97	.0028	-.0042	.0011	-2	11
13	926	1	23	3	36	-.04	-.65	.76	.0023	-.0027	-.0022	4	-1
15	938	1	24	3	43	-.03	-.90	.43	.0016	-.0009	-.0018	2	0
17	961	1	26	1	38	.00	.85	.53	.0019	-.0008	.0013	1	-1
20	1408	3	4	9	23	.00	.57	.82	.0014	.0014	-.0010	6	-42
21	1479	3	10	7	42	.01	.21	.98	.0014	.0022	-.0005	5	-51
23	1681	3	27	4	48	.00	.73	.68	.0013	.0009	-.0010	16	14
25	1752	4	2	3	7	.00	.52	.85	.0014	.0016	-.0010	18	15

TABLE 4

EQUATORIAL ARRAY. Measurement Errors.

ARC No.	Standard Deviation Of Measurement Error About Mean For Pass		Mean Measurement Error For Pass		MINITRACK COUNTS	
	σl	σm	Δl	Δm	Δl	Δm
1	.000191	.000213	.000151	-.000080	7	-4
2	.000081	.000053	-.000013	-.000050	-1	-2
4	.000033	.000040	.000062	.000025	3	1
13	.000189	*	.000157	*	7	*
15	.000657	.000757	.000192	-.000397	9	-18
17	.000088	.000076	.000056	.000006	3	0
20	*	.000041	*	-.000007	*	0
21	.000031	.000041	-.000054	-.000040	-2	-2
23	.000071	.000043	.000113	.000012	5	1
25	.000034	.000039	-.000045	.000038	-2	2

EW - BIAS: 957 MINICTS + .000026 = 958 MINICTS

NS - BIAS: 988 MINICTS - .000544 = 988 MINICTS

NS - SCALE FACTOR: -.00049

(Base line is shorter than assumed.)

TABLE 5

COMPARISON OF CALIBRATION RESULTS

Source	DBA*	RCA**	Aircraft Calibration July 1965	Aircraft Calibration 5 Nov 1965	Aircraft Calibration 3 Mar. 1966	Actual Values Used 5 Nov 1965	Actual Values Used 3 Mar. 1966
Equatorial Array							
EW - BIAS	958	956	956	957	969	954	959
NS - BIAS	988	985	988	992	988	988	988
Polar Array							
EW - BIAS	136	136	137	140	153	138	143
NS - BIAS	435	437	431	435	433	435	433
Equatorial Array							
NS - Scale Factor	-.00049	-.00015					
Polar Array							
EW - Scale Factor	-.00022	-.00022					

* The DBA computed calibration values are the ones shown in Tables I and II. They are based on observations from 4 January to 2 April 1966.

** The RCA results have been verbally communicated by Mr. Jerry Casto and are based on satellite observations covering approximately the same time period as those used by DBA.

TABLE 6

OBSERVED GEOS - A FREQUENCIES

(Source: Harry Pritchard, GSFC)

Date (1966)	Value (MHz)	Scale Error Relative To Nominal Value (136.83 MHz)
1 January	136.830360	26×10^{-7}
5 January	136.830350	26×10^{-7}
3 February	136.830038	3×10^{-7}
8 February	136.829675	-24×10^{-7}
17 March	136.830540	40×10^{-7}

TABLE 7

CHANGE IN PREDICTED MEASUREMENTS DUE TO INCLUSION OF THE DOPPLER EFFECT AND WAVE PROPAGATION TIME DELAY

POLAR ARRAY

ARC No.	$10^6 \Delta l$	$10^6 \Delta m$
1	-14	19
2	-14	20
3	-10	21
5	-12	21
7	-11	21
9	-11	21
10	-14	18
11	-10	21
12	-12	20
15	-11	21
16	-14	18
17	-14	19
18	-13	20
19	-11	20
20	x	-18
21	-11	-18
22	- 8	-20
23	-12	-17
24	-11	-18
25	-11	-17

EQUATORIAL ARRAY

ARC No.	$10^6 \Delta l$	$10^6 \Delta m$
1	-15	17
2	-15	18
4	-13	20
13	-11	x
15	-11	21
17	-14	18
20	x	-17
21	-11	-18
23	-13	-16
25	-12	-17

POLAR ARRAY: 1 MINITRACK Count = .0000175

EQUATORIAL ARRAY: 1 MINITRACK Count = .0000216

2.5.2 The Calibration of all Minitrack Stations Using GEOS-A Minitrack Measurements

The procedures followed in processing the Minitrack data are described in Section 2.6 of this report. The four arcs described in Table 15 were at first processed separately. A new data tape was created for each of the four arcs (see special subroutines PRTIAL and RESID). The arcs were then combined in pairs to give two, two-arc runs, and finally all the data were combined into a single four-arc run. Three sets of runs were made under slightly varying conditions (see Table 15): In the first run scale factors and antenna rotation terms were recovered in addition to measurement biases and the four state vectors. The second run was similar to the first except that no rotation terms were considered. This resulted in very slight differences in the solutions. The third run was similar to the first but in this run ionospheric corrections were made using the ionospheric model developed by DBA under Contract NAS511730. In all runs, data were edited out of the solution using the criteria listed in Tables 12 and 13. It should be mentioned that when the runs were made, no ionospheric data could be obtained for 1966. The 1971 values were therefore substituted. This may account for the relatively bad results obtained for run 3. Tables 8 through 14 refer to run 1. Table 8 is a comparison of calibration results. Table 9 lists the scale factor errors and antenna alignment errors. It can be seen that the scale errors are predominantly negative. This is the kind of effect that would be expected if no ionospheric corrections were made, i.e., it seems likely that the scale factors have absorbed some of the effect of ionospheric refraction. When ionospheric corrections were made (run 3), the recovered scale factors were indeed reduced in magnitude (the actual numbers are not included in this report.

Table 10 is a listing of the number of observations for each station. Table 11 gives the RMS values of the observed residuals. It is interesting to note that the Polar Array, which has a larger baseline, shows a larger RMS error than the Equatorial Array when the errors are expressed in Minitrack counts, but more or less the same RMS error in terms of angular error. From this, it could be inferred that the errors are caused less by the electronics of the observing stations, which measures Minitrack counts, than by general uncertainties in conditions outside the observing station. This, in general, is probably true. However, it has been observed that large variations in the precalibration biases, which is part of the Minitrack message, have been accompanied by large residuals in the orbital fit. The Woomera Polar Station was particularly bad in this report. The measurement residuals and precalibration constants for that station are listed in Table 14 for illustrative purposes. Tables 12 and 13 are self-explanatory, as are Tables 15.1 through 15.4. It should be emphasized that the results given in Tables 15 are the most important results of this report.

In all orbital computations the Earth model used was that given in Appendix A-1. The affect of solar radiation pressure was not taken into account, and neither was the effect of the Earth's precession and nutation.

In all three runs the measurements were weighted proportionately to the standard deviation of the measurements within each message.

TABLE 8

Comparison of Calibration Results for

Equatorial and Polar Station Biases

Station	This Program *		Aircraft 65.12.31		Aircraft 66.02.25	
	EW	NS	EW	NS	EW	NS
Fort Myers (E)	956	981	954	988	953	984
Fort Myers (P)	135	435	138	435	137	435
Quito (E)	817	036	812	036	837	030
Quito (P)	756	000	760	002	763	006
Lima (E)	949	252	955	263	955	259
Lima (P)	023	881	028	882	028	882
Santiago (E)	078	057	073	060	076	059
Santiago (P)	950	972	943	967	943	963
New Foundland (E)	930	937	934	935	934	935
New Foundland (P)	170	920	171	919	171	919
Winkfield (E)	071	973	067	976	067	976
Winkfield (P)	868	001	870	011	870	011
Johannesburg (E)	048	937	054	932	054	932
Johannesburg (P)	084	876	090	871	090	871
Blossom Point (E)	948	094	948	096	948	096
Blossom Point (P)	081	887	082	889	082	889
College (E)	955	807	961	804	960	804
College (P)	846	932	854	916	859	919
Mojave (E)	030	576	017	579	029	575
Mojave (P)	079	123	062	129	079	121
Grand Forks (E)	068	951	060	954	061	951
Grand Forks (P)	602	001	598	990	598	990
Woomera (E)	020	944	031	938	031	938
Woomera (P)	934	118	948	118	948	118

* Time span covered was 66.1.1 - 66.2.3 and 66.3.12 - 66.3.19.

Table 9

Scale Factor Errors and Antenna Alignment Errors in Minitrack Counts

1000 Equatorial Minitrack Counts = .0216 radians
1000 Polar Minitrack Counts = .0174 radians

	Equatorial Array		Polar Array	
	Scale Error	Antenna Rotation	Scale Error	Antenna Rotation
Fort Myers	-14	4	-15	2
Quito	9	-4	- 6	-2
Lima	-24	-4	30	-3
Santiago	- 2	-8	- 8	-2
New Foundland	- 8	0	-13	-2
Winkfield	- 4	-3	- 8	5
Johannesburg	- 6	-1	-14	-4
Blossom Point	-11	6	- 1	0
College	-10	8	- 4	0
Mojave	- 3	-3	- 7	4
Grand Forks	8	-5	- 8	-9
Woomera	-11	-3	- 8	-4

Scale Error — The predicted measurement is in error by an amount equal to the "scale error" multiplied by the actual measurement.

Antenna Rotation — The predicted measurement is in error because the antenna baselines are rotated anti-clockwise by the "antenna rotation".

In the equatorial mode the scale error only affects the m measurement and the antenna rotation only the l measurement.

In the polar mode the scale error only affects the l measurement and the antenna rotation only the m measurement.

Table 10

The Number of Minitrack Messages Used in the Orbit Determination and the Determination of Biases, Scale Factors and Antenna Rotations

Equatorial Array	Direction Cosine l					Direction Cosine m				
	Arc 1	Arc 2	Arc 3	Arc 4	Total	Arc 1	Arc 2	Arc 3	Arc 4	Total
Fort Myers	4	4	11	12	31	4	4	11	12	31
Quito	1	-	2	4	7	1	-	-	3	4
Lima	-	3	-	2	5	-	3	2	-	5
Santiago	1	1	3	3	8	1	1	3	3	8
New Foundland	17	19	26	31	93	16	19	26	31	92
Winkfield	12	-	9	11	32	12	-	15	11	38
Johannesburg	-	1	3	2	6	-	1	3	2	6
Blossom Point	12	9	20	16	57	12	9	21	16	58
College	12	12	18	17	59	13	12	17	17	59
Mojave	12	11	17	14	54	12	11	17	14	54
Grand Forks	11	14	22	7	54	12	14	22	7	55
Woomera	1	-	1	3	5	-	-	2	3	5
Total	83	74	132	122	411	83	74	139	119	415

Table 10 (Continued)

The Number of Minitrack Messages Used in the Orbit Determination and the Determination of Biases, Scale Factors and Antenna Rotations

Polar Array	Direction Cosine l					Direction Cosine m				
	Arc 1	Arc 2	Arc 3	Arc 4	Total	Arc 1	Arc 2	Arc 3	Arc 4	Total
Fort Myers	11	9	14	19	53	10	9	14	19	52
Quito	4	3	5	5	17	4	4	5	5	18
Lima	5	5	5	4	19	5	6	6	6	21
Santiago	4	2	5	5	16	4	3	5	4	16
New Foundland	9	14	14	17	54	9	14	14	18	55
Winkfield	3	-	2	3	8	3	-	2	2	7
Johannesburg	2	3	7	7	19	2	3	7	8	20
Blossom Point	13	9	15	20	57	12	9	15	20	58
College	7	5	9	8	29	7	6	9	7	29
Mojave	14	11	13	20	58	14	11	13	20	58
Grand Forks	4	7	18	10	39	4	7	18	10	39
Woomera	4	2	3	7	16	3	3	5	7	18
Total	80	71	110	125	385	77	75	112	124	389

Table 11

RMS Value of Residuals* in Minitrack Counts

Station	Equatorial Array l	Equatorial Array m	Polar Array l	Polar Array m
Fort Myers	2.7	5.6	5.1	4.5
Quito	3.2	8.9	7.4	5.2
Lima	3.7	9.3	6.6	5.7
Santiago	4.7	3.8	7.2	7.0
New Foundland	4.9	3.3	7.1	5.3
Winkfield	7.3	3.5	4.5	5.4
Johannesburg	3.6	2.8	4.5	4.0
Blossom Point	4.8	3.4	4.8	3.2
College	3.7	5.9	5.8	8.4
Mojave	3.1	5.3	5.2	5.1
Grand Forks	4.9	4.2	6.0	7.1
Woomera	4.1	11.8	7.2	8.1
All **	4.3	4.7	5.9	5.6

*The residuals are the differences between observed and predicted values. The predicted values are based on the computed orbit.

**It is of interest to note that the four RMS residuals for all stations expressed as unscaled quantities assume the values:

.000093, .000102, .000102, .00097.

The same residuals expressed as equivalent angular discrepancies at zenith assume the values: 19", 21", 21", 20", i.e., values around the quoted Minitrack accuracy of 20".

Table 12

Total Number of Minitrack Messages and the Number of Rejected Messages

Equatorial Array	Total Number Messages	Rejected (Elevation)	Direction Cosine l Rejected (Discrepancy)	Direction Cosine l Rejected (S.D.)	Direction Cosine m Rejected (Discrepancy)	Direction Cosine m Rejected (S.D.)
Fort Myers	52	20		1		1
Quito	7				3	
Lima	5					
Santiago	9		1			1
New Foundland	99	5	1		3	
Winkfield	41		9		3	
Johannesburg	6					
Blossom Point	73	15	1			
College	69	6	3	1	3	1
Mojave	79	25				
Grand Forks	59	4	1			
Woomera	6		1		1	
Total	505	75	17	2	12	3

Rejection Criteria

Elevation. All messages the mean elevation of which was below 25°.

Discrepancy. All direction cosines having a residual error in excess of 20 Minitrack Counts.

S.D. All direction cosines having an error with a standard deviation in excess of 0.00050, the computed standard deviation being based on all measurements within a Minitrack message.

Rejected measurements are only listed under one heading even if they should have been rejected for more than one reason. Thus, the total number of measurements less the number of rejected measurements equals the number of measurements used in the computations (Table 3).

Table 12 (Continued)

Total Number of Minitrack Messages

and the Number of Rejected Messages

Polar Array	Total Number Messages	Rejected Elevation	Direction Cosine l		Direction Cosine m	
			Rejected Discrepancy	Rejected (S.D.)	Rejected Discrepancy	Rejected (S.D.)
Fort Myers	72	18		1		2
Quito	19		1	1	1	
Lima	21			2		
Santiago	22	1	4	1	3	2
New Foundland	74	19	1			
Winkfield	8				1	
Johannesburg	20		1			
Blossom Point	82	24	1		2	
College	76	40	1	6	1	6
Mojave	70	10	1	1	1	1
Grand Forks	51	11	1			1
Woomera	20		4		2	
Total	535	123	15	12	11	12

Rejection Criteria

Elevation. All messages the mean elevation of which was below 25°.

Discrepancy. All direction cosines having a residual error in excess of 20 Minitrack Counts.

S.D. All direction cosines having an error with a standard deviation in excess of 0.00050, the computed standard deviation being based on all measurements within a Minitrack message.

Rejected measurements are only listed under one heading even if they should have been rejected for more than one reason. Thus, the total number of measurements less the number of rejected measurements equals the number of measurements used in the computations (Table 3).

Table 13

Total Number of Measurements and Number of Rejected Messages

	Total Measurements	Rejected (Elevation)	Rejected (Discrepancy)	Rejected (S.D.)	Used In Computation
Number	2080	396	55	29	1600
Percent	100	19.0	2.6	1.4	77.0

Rejection Criteria

Elevation.	All measurements at a mean elevation below 25°.
Discrepancy	All measurements having a residual error in excess of 20 Minitrack Counts
S.D.	All measurements having an error with a standard deviation in excess of 0.00050, the computed standard deviation being based on all measurements within a Minitrack message.

Rejected measurements are only listed under one heading even if they should have been rejected for more than one reason.

Table 14

Residual Measurement Errors and Prepass Calibration Constants

for the Woomera Polar Station

Arc	Date	Time (hrs)	Residual Measurement Error l	m	KC-KS1 + KS2 l	m
1	1.2.66	5.2	11	3	345	557
1	1.3.66	18.7	-10	5	344	545
1	1.4.66	18.8	- 2	30	343	551
1	1.5.66	18.8	0	5	342	570
2	1.11.66	17.2	- 9	-14	345	584
2	1.12.66	17.2	5	- 8	343	584
2	1.13.66	17.3	31	-16	305	576
3	1.28.66	14.2	65	7	152	546
3	1.28.66	22.8	98	- 2	124	544
3	1.29.66	14.3	-34	65	224	545
3	1.29.66	22.9	-16	12	222	541
3	1.30.66	22.9	3	4	221	540
3	2.3.66	12.6	14	10	160	541
4	3.12.66	4.8	1	-11	561	719
4	3.12.66	13.4	- 4	0	561	717
4	3.13.66	4.9	0	- 1	560	717
4	3.13.66	13.5	4	- 5	560	717
4	3.15.66	13.6	0	- 1	560	717
4	3.16.66	13.7	- 2	1	561	713
4	3.19.66	3.3	2	-12	563	716

Table 15.1

Comparison of a GEOS-A Arc Determined from Minitrack Data with an Orbit Determined from Optical Data

Start Time	1.1.66	6 hours
Stop Time	1.5.66	19 hours
Length of Arc	4.5 days	

	Run 1	Run 2	Run 3
RMS Position Difference (m)	67	69	149
RMS Velocity Difference (m/sec)	.044	.046	0.100
Maximum Position Difference (m)	103	122	237
Maximum Velocity Difference (m/sec)	.063	.074	0.140

Run 1. Minitrack orbit with measurement bias, scale factor, and rotation recovery. No ionospheric corrections.

Run 2. Minitrack orbit with measurement bias and scale factor recovery. No ionospheric corrections.

Run 3. Minitrack orbit with measurement bias, scale factor, and rotation recovery. Ionospheric corrections.

It is of interest to compare the above results with those quoted in "Intercomparison of the Minitrack and Optical Tracking Networks using GEOS-I Long Arc Orbital Solutions" (by J.G.Marsh, C.E.Doll, R.J.Sandifer and W.A.Taylor, NASA-TND-5337, February 1970). For an arc covering almost the identical period, they obtained an RMS position difference of 165 meters in a similar comparison. However, they did not attempt to recover any station calibration parameters. They made no ionospheric corrections.

Table 15.2

Comparison of a GEOS-A Arc Determined from Minitrack Data with an Orbit Determined from Optical Data

Start Time	1.8.66 0 hours
Stop Time	1.14.66 5 hours
Length of Arc	6.2 days

	Run 1	Run 2	Run 3
RMS Position Difference (m)	86	116	160
RMS Velocity Difference (m/sec)	.075	.099	.127
Maximum Position Difference (m)	127	180	262
Maximum Velocity Difference (m/sec)	.120	.158	.180

Run 1. Minitrack orbit with measurement bias, scale factor, and rotation recovery. No ionospheric corrections.

Run 2. Minitrack orbit with measurement bias and scale factor recovery. No ionospheric corrections.

Run 3. Minitrack orbit with measurement bias, scale factor, and rotation recovery. Ionospheric corrections.

Table 15.3

Comparison of a GEOS-A Arc Determined from Minitrack Data with an Orbit Determined from Optical Data

Start Time	1.28.66	2 hours
Stop Time	2.4.66	0 hours
Length of Arc	6.9 days	

	Run 1	Run 2	Run 3
RMS Position Difference (m)	124	114	182
RMS Velocity Difference (m/sec)	.103	.099	.155
Maximum Position Difference (m)	194	181	272
Maximum Velocity Difference (m/sec)	.155	.153	.231

Run 1. Minitrack orbit with measurement bias, scale factor, and rotation recovery. No ionospheric corrections.

Run 2. Minitrack orbit with measurement bias and scale factor recovery. No ionospheric corrections.

Run 3. Minitrack orbit with measurement bias, scale factor, and rotation recovery. Ionospheric corrections.

Table 15.4

Comparison of a GEOS-A Arc Determined from Minitrack Data with an Orbit Determined from Optical Data

Start Time	3.12.66	3 hours
Stop Time	3.19.66	20 hours
Length of Arc	7.6 days	

	Run 1	Run 2	Run 3
RMS Position Difference (m)	168	179	249
RMS Velocity Difference (m/sec)	.142	.152	.207
Maximum Position Difference (m)	251	250	406
Maximum Velocity Difference (m /sec)	.207	.215	.315

Run 1. Minitrack orbit with measurement bias, scale factor, and rotation recovery. No ionospheric corrections.

Run 2. Minitrack orbit with measurement bias and scale factor recovery. No ionospheric corrections.

Run 3. Minitrack orbit with measurement bias, scale factor, and rotation recovery. Ionospheric corrections.

2.6 RECOMMENDED PROCEDURE FOR REDUCING MINITRACK DATA

2.6.1 Reduction Procedure

The purpose of this section is to provide a step-by-step procedure to be followed for the reduction of Minitrack data and optical data. This procedure can be simplified into three or four parts.

PREPROCESS MINITRACK DATA → NAP-II PROGRAM REDUCTION ← PREPROCESS OPTICAL DATA

NAP-II PROGRAM REDUCTION → REPORT RESULTS

The preprocessing of the Minitrack data and the Optical data can be done as a series or parallel effort. The same applies to the NAP-II reduction of the data.

The steps to a complete reduction of the Minitrack data are:

1. Extract desired data from Minitrack and/or Optical data tapes.
2. Sort Minitrack messages in time sequence (not required for Optical data when GEOS formatted data tapes are used).
3. Preprocess Minitrack and/or Optical data.
4. Set up NAP-II control cards and write cards on magnetic tape.
5. Make PRENAP card updater run.
6. NAP-II program execution.
7. POSTNAP program execution.

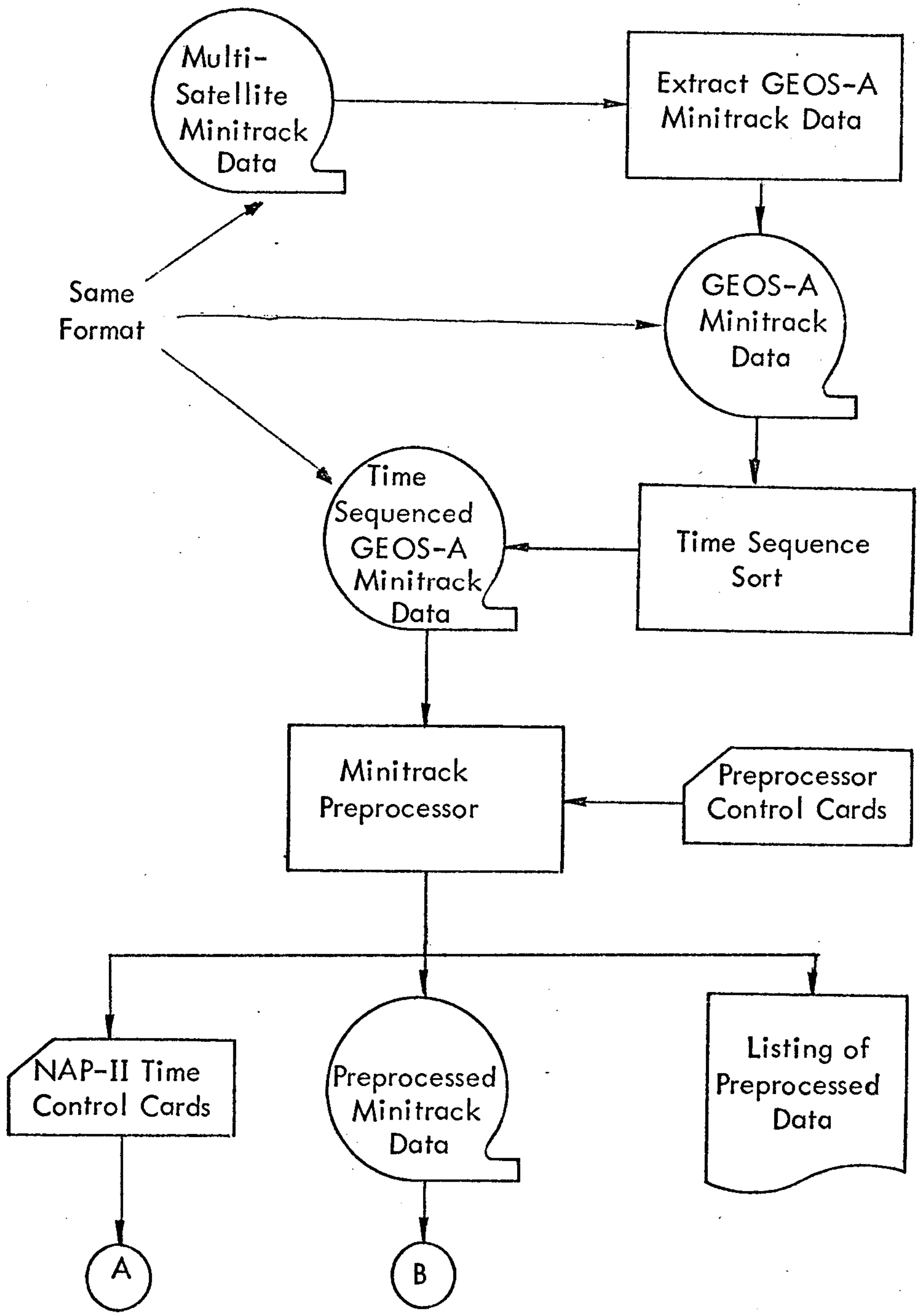
   Iterate on these 3 steps until convergence achieved

8. Report results after reduction has converged.

A flow diagram for the preprocessing of the Minitrack data is given in Figure 1 . Note: In setting up these runs, refer to Appendix A-4 for restriction place on card set-up.

FIGURE 1

MINITRACK PREPROCESSING

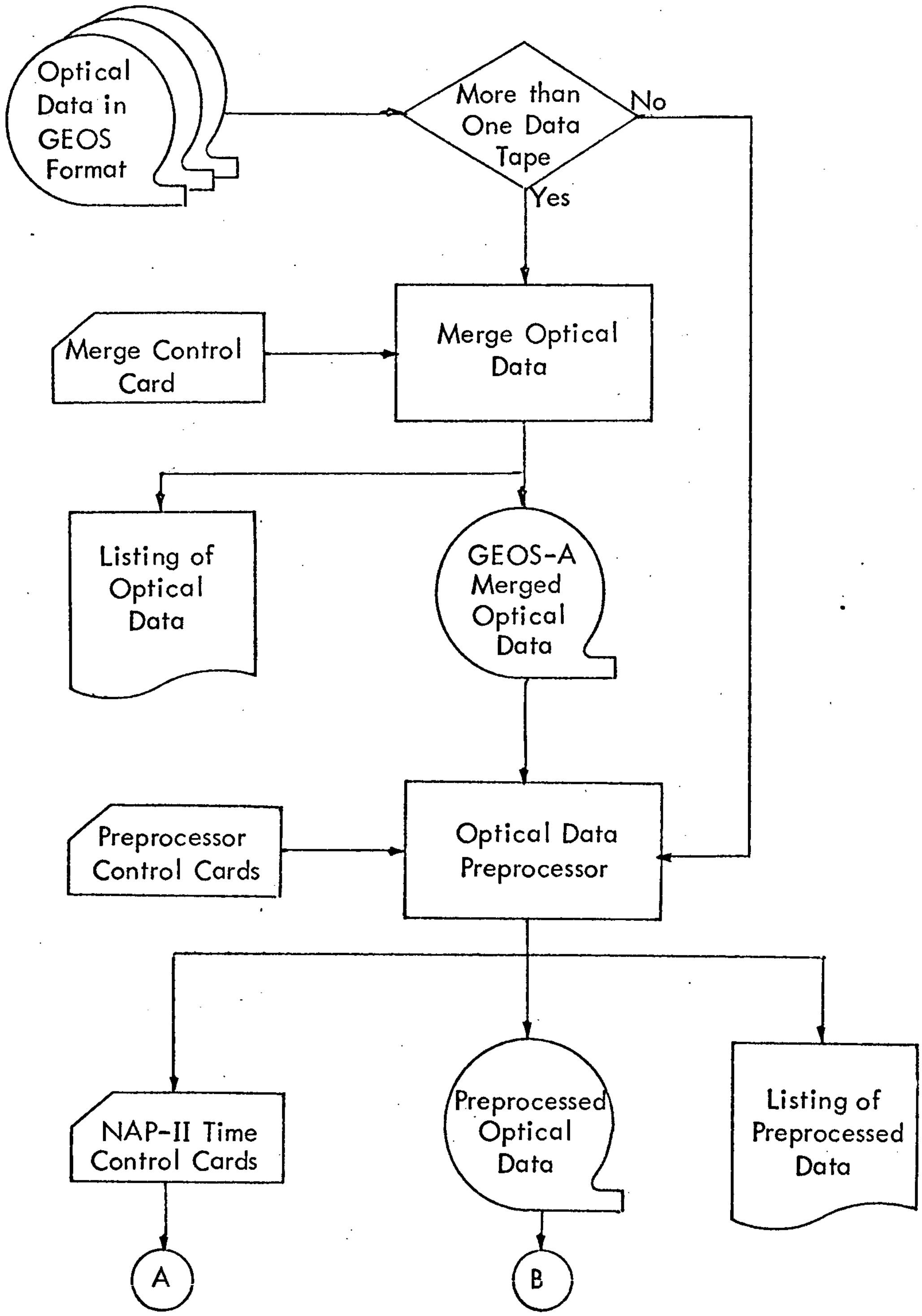


The process starts with a magnetic tape containing Minitrack messages from several satellites. This tape is input to a program to extract just the messages from GEOS-1 satellite and writes them on another magnetic tape in the same format. This tape is then input to a program that sorts the messages, and also edits duplicate messages and puts them in time sequence on magnetic tape. This tape is now in the right form for the preprocessor program. The time-sequenced data tape, along with preprocessor control cards, are now input to the Minitrack preprocessor program. The preprocessor makes known corrections to the data and converts the corrected phase differences to direction cosines. The program outputs a data tape in a format acceptable to NAP-II and also a listing of the corrected data. The preprocessor also outputs a bulk of the NAP-II control cards dealing with the station times for the data (Category 201, 202, and 999). These are used in selection of data to be processed. This ends the preprocessing phase of the procedure. The preprocessing of Minitrack data is discussed in Section 2.2 of this report.

Figure 2 is a flow diagram of the preprocessing of optical data. The data input to the Optical preprocessing program must be in time sequence. If there are more than one tracking systems data to be processed, the data tapes must be "merged" onto one data tape. The output from the merge program is a data tape, in GEOS format, and a listing of the data. This data tape is input to the Optical preprocessor with control cards that specify the stations and times of the data to be processed. In the case of the SAO stations, these cards are also used to correct the observation times from A.1 to UTC. The Optical preprocessor outputs a data tape in a format acceptable to NAP-II and punched cards used in NAP-II to control data times. There is also a printed output of the reformatted and corrected data. The optical data preprocessing is discussed in Section 2.3 of this report.

FIGURE 2

OPTICAL PREPROCESSING



Once the data has been preprocessed, it is ready to be reduced using the NAP-II program. This process is flowed in Figure 3. The first step is to set up the NAP-II control cards as specified by the NAP-II User's Guide. The timing cards (Category 201, 202, and 999) are output as a result of the preprocessor. After the cards are set up, they are written on magnetic tape, for ease of handling. The cards on tape can be changed via the PRENAP card updater. This program is discussed in Section 2.4.1 of this report.

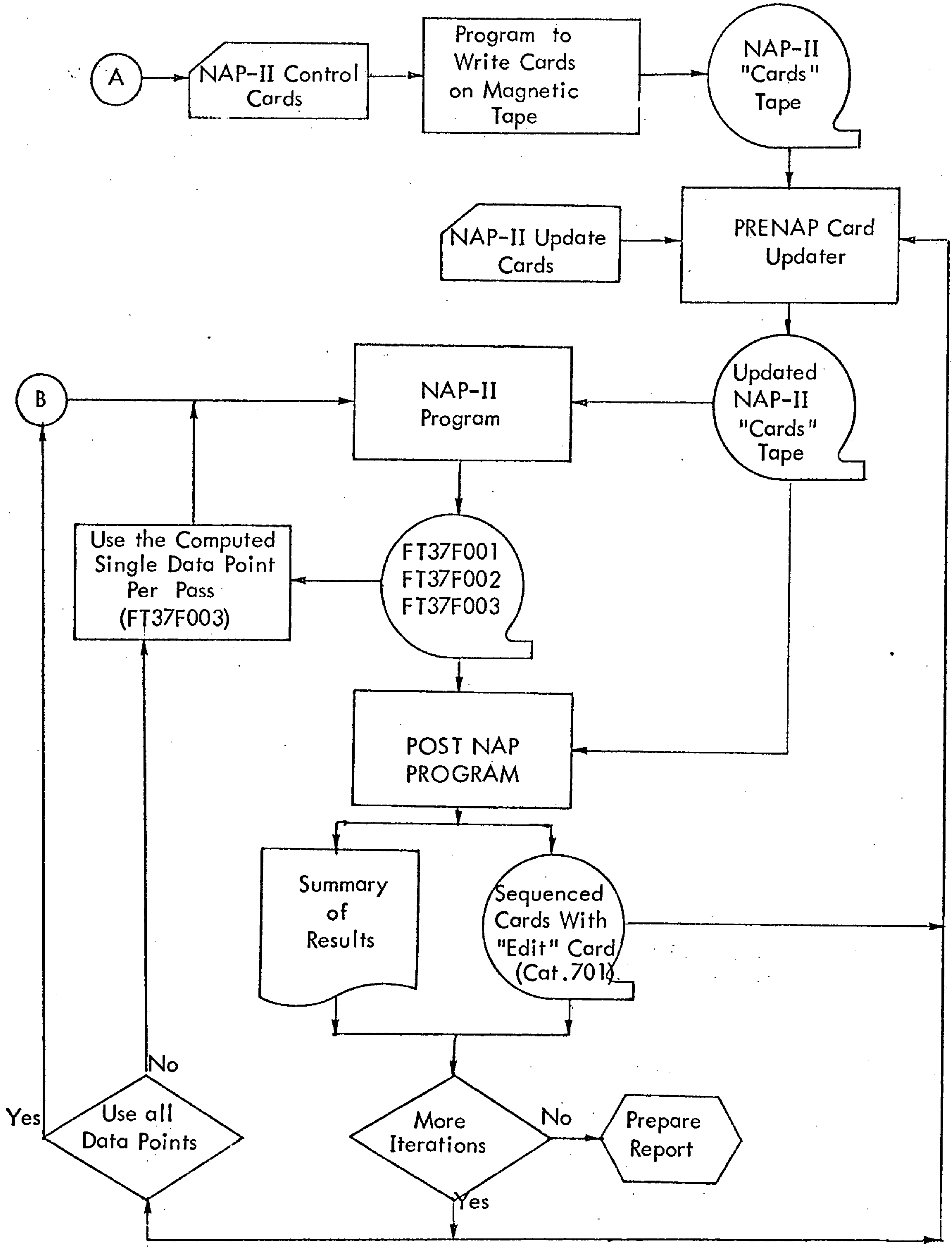
The NAP-II control cards (now on tape) and the data tape output from the preprocessor are now input to the NAP-II program. At the end of each iteration, NAP-II outputs a disk file (on tape) with the current values of the error model parameters and the measurement discrepancies. Also output is a "single point" data tape which can be used on successive iterations to reduce the computation time. When using the "single point" data tape for Minitrack data, delete the Category 704 cards.

The output from the NAP-II program is then input to the POSTNAP program which, in effect, gives a status report of the reduction and prepares information for the next iteration.

The PRENAP, NAP, POSTNAP sequence is continued until the solution has converged.

FIGURE 3

NAP-II REDUCTION



C.5

2.6.2 Recommended Error Model

The results of this study indicate the following error models to be sufficient for reducing Minitrack data.

The error model terms to be used are dependent on whether the data was collected in the equatorial or polar modes.

EQUATORIAL ARRAY

$$\ell = \ell_0 + b_1 + r_e m_0 + \tau\dot{\ell}$$

$$m = m_0 + b_2 + s_e m_0 + \tau\dot{m}$$

POLAR ARRAY

$$\ell = \ell_0 + b_3 + s_p \ell_0 + \tau\dot{\ell}$$

$$m = m_0 + b_4 + r_p \ell_0 + \tau\dot{m}$$

where,

ℓ_0 and m_0 are Minitrack measurements in Minitrack counts/1000.

b_1, b_2, b_3 and b_4 are zero-set biases

r_e and r_p are rotation terms about the local vertical of equatorial ℓ and polar m measurements respectively

s_e and s_p are scale factors of equatorial m and polar ℓ measurements respectively

τ is the timing error associated with the station

$\dot{\ell}$ and $\dot{m}$ are computed rates obtained from the orbital equations within NAP-II

These error model terms are coded for input to NAP-II (see Reference 4, The NAP-II User's Guide, Appendix IV-B).

The error model terms would correspond to the following term numbers for direction cosine data types (measurement code 6 and 7).

Error Model Term	Error Model Term No.
b_1, b_2, b_3, b_4	10
r_e and r_p	15 (see changes to MESOLD in Section 2.4.2)
s_e and s_p	14
τ	11

Note: Consult NAP-II User's Guide, Appendix IV-2.

2.6.3 Typical Set-up for NAP-II Program

The following is an example of a set of NAP-II control cards. This listing is output from the NAP-II program on each run.

Note that each station is defined twice (category 301, 302, and 303); once for the equatorial array and once for the polar array. Following the survey cards are the error model cards (category 601). For ease of recognition, the labels have been coded as to which term it is, i.e. FTMYRELB is the FTMYR station equatorial "ℓ" measurement zero-set bias "B". FTMYRPLB is the polar "ℓ" measurement bias. Other codes used are "ELS" and "PLS" for equatorial "ℓ" scale and polar "ℓ" scale, "ELR" and "PLR" for equatorial "ℓ" rotation and polar "ℓ" rotation.

It is recommended that a new user of NAP-II take the following listing and, using the NAP-II User's Guide, cross-reference the codes used in the listing to their meaning given in the User's Guide.

See Appendix A-4 for restriction on deck setup.

CONTROL DATA LISTING

CATEGORY	CONTINUE	LABEL	KEY1	KEY2	KEY3	KEY4	KEY5	KEY6	KEY7	KEY8	KEY9	KEY10	DATA1	DATA2
101	0	MINARC2	2	1	1	0	2	2	0	1	0	1	0.0	0.0
102	0	LCCNTRCY	5	0	0	0	0	0	0	0	0	0	0.66944970381233990000D-02	0.0
103	6		2	13	12	0	0	0	0	0	0	0	-0.12629900000000000000D-18	0.0
103	7		2	13	12	0	0	0	0	0	0	0	0.16522000000000000000D-18	0.0
103	6		2	14	12	0	0	0	0	0	0	0	0.13597800000000000000D-20	0.0
103	7		2	14	12	0	0	0	0	0	0	0	-0.13177200000000000000D-19	0.0
103	6		2	15	12	0	0	0	0	0	0	0	-0.13212600000000000000D-19	0.0
103	7		2	15	12	0	0	0	0	0	0	0	-0.12963900000000000000D-20	0.0
103	7		2	9	1	0	0	0	0	0	0	0	0.78400000000000000000D-03	0.0
103	6		2	12	12	0	0	0	0	0	0	0	-0.27830000000000000000D-18	0.0
103	7		2	12	12	0	0	0	0	0	0	0	0.71800000000000000000D-20	0.0
103	7		2	13	12	0	0	0	0	0	0	0	0.11652000000000000000D-18	0.0
103	6		2	13	13	0	0	0	0	0	0	0	-0.23906000000000000000D-19	0.0
103	7		2	13	13	0	0	0	0	0	0	0	0.21160000000000000000D-19	0.0
103	6		2	14	11	0	0	0	0	0	0	0	0.54659999999999990000D-21	0.0
103	7		2	14	11	0	0	0	0	0	0	0	-0.47299999999999990000D-21	0.0
103	6		2	14	13	0	0	0	0	0	0	0	0.57000000000000000000D-21	0.0
103	7		2	14	13	0	0	0	0	0	0	0	0.66000000000000000000D-20	0.0
103	6		2	14	14	0	0	0	0	0	0	0	-0.19310000000000000000D-21	0.0
103	7		2	14	14	0	0	0	0	0	0	0	-0.41390000000000000000D-22	0.0
103	6		2	15	9	0	0	0	0	0	0	0	-0.24100000000000000000D-18	0.0
103	7		2	15	9	0	0	0	0	0	0	0	-0.48300000000000000000D-18	0.0
103	6		2	15	13	0	0	0	0	0	0	0	-0.77009999999999990000D-21	0.0
103	7		2	15	13	0	0	0	0	0	0	0	-0.37359999999999990000D-21	0.0
103	6		2	15	14	0	0	0	0	0	0	0	0.11400000000000000000D-22	0.0
103	7		2	15	14	0	0	0	0	0	0	0	-0.55799999999999990000D-22	0.0
104	0	INTEGRAT	4	0	3	4	1	2	0	0	0	0	0.21600000000000000000D 05	0.50000000000000000D-12
104	1	EPOCH 01	0	1	0	0	0	68	2	27	23	56	0.56400000000000000000D 02	0.0
104	4	SUN X	4	1	0	0	0	0	0	0	0	0	0.13783915359200000000D 12	0.0
104	4	SUN Y	4	2	0	0	0	0	0	0	0	0	-0.49588863289899990000D 11	0.0
104	4	SUN Z	4	3	0	0	0	0	0	0	0	0	-0.21636861132300000000D 11	0.0
104	4	SUN XD	4	4	0	0	0	0	0	0	0	0	0.11424768869600000000D 05	0.0
104	4	SUN YD	4	5	0	0	0	0	0	0	0	0	0.25530416236700000000D 05	0.0
104	4	SUN ZD	4	6	0	0	0	0	0	0	0	0	0.11073121988100000000D 05	0.0
104	4	MOON X	3	1	0	0	0	0	0	0	0	0	0.38152909646700000000D 09	0.0
104	4	MOON Y	3	2	0	0	0	0	0	0	0	0	-0.14962277913100000000D 09	0.0
104	4	MOON Z	3	3	0	0	0	0	0	0	0	0	-0.87006519876999900000D 08	0.0
104	4	MOON XD	3	4	0	0	0	0	0	0	0	0	0.46598105611600000000D 03	0.0
104	4	MOON YD	3	5	0	0	0	0	0	0	0	0	0.80053902731400000000D 03	0.0
104	4	MOON ZD	3	6	0	0	0	0	0	0	0	0	0.41547916292600000000D 03	0.0
301	0	FTMPQ	1	5	0	3	0	0	0	0	26	32	0.53219000000000000000D 02	0.0
302	0	FTMEQ	1	0	0	0	0	0	0	0	278	8	0.37400000000000000000D 01	0.0
30	0	FTMEQ	1	0	0	0	0	0	0	0	0	0	-0.43681000000000000000D 02	0.0
301	0	FTMPC	2	5	0	103	0	0	0	0	26	32	0.53219000000000000000D 02	0.0
302	0	FTMPD	2	0	0	0	0	0	0	0	278	8	0.37400000000000000000D 01	0.0
303	0	FTMPG	2	0	0	0	0	0	0	0	0	0	-0.43681000000000000000D 02	0.0
301	0	OUITOGE	3	5	0	5	0	0	0	-1	0	37	0.22758000000000000000D 02	0.0
302	0	OUITOGE	3	0	0	0	0	0	0	0	281	25	0.15349000000000000000D 02	0.0
303	0	OUITOGE	3	0	0	0	0	0	0	0	0	0	0.35578010000000000000D 04	0.0
301	0	OUITOOP	4	5	0	105	0	0	0	-1	0	37	0.22796000000000000000D 02	0.0

CONTROL DATA LISTING

CATEGORY	CONTINUE	LABEL	KEY1	KEY2	KEY3	KEY4	KEY5	KEY6	KEY7	KEY8	KEY9	KEY10	DATA1	DATA2
302	0	QUITQ5P	4	0	0	0	0	0	0	0	281	25	0.15345000000000000000D 02	0.0
303	0	QUITQ6P	4	0	0	0	0	0	0	0	0	0	0.35578010000000000000D 04	0.0
301	0	LIMAP5E	5	5	0	6	0	0	0	-1	11	40	0.37620000000000000000D 02	0.0
302	0	LIMAP6E	5	0	0	0	0	0	0	0	232	50	0.58621000000000000000D 02	0.0
303	0	LIMAP6E	5	0	0	0	0	0	0	0	0	0	0.57279000000000000000D 02	0.0
301	0	LIMAP6P	6	5	0	100	0	0	0	-1	11	40	0.37620000000000000000D 02	0.0
302	0	LIMAP5P	6	0	0	0	0	0	0	0	282	50	0.58621000000000000000D 02	0.0
303	0	LIMAP6P	6	0	0	0	0	0	0	0	0	0	0.57279000000000000000D 02	0.0
301	0	SNTAG6E	7	5	0	8	0	0	0	-1	33	8	0.59003000000000000000D 02	0.0
302	0	SNTAG6E	7	0	0	0	0	0	0	0	289	19	0.52609000000000000000D 02	0.0
303	0	SNTAG6E	7	0	0	0	0	0	0	0	0	0	0.73254000000000000000D 03	0.0
301	0	SNTAG6P	8	5	0	108	0	0	0	-1	33	8	0.59003000000000000000D 02	0.0
302	0	SNTAG6P	8	0	0	0	0	0	0	0	289	19	0.52609000000000000000D 02	0.0
303	0	SNTAG6P	8	0	0	0	0	0	0	0	0	0	0.73254000000000000000D 03	0.0
301	0	NEWFL6E	9	5	0	12	0	0	0	0	47	44	0.29000000000000000000D 02	0.0
302	0	NEWFL6E	9	0	0	0	0	0	0	0	307	15	0.46862000000000000000D 02	0.0
303	0	NEWFL6E	9	0	0	0	0	0	0	0	0	0	0.52328000000000000000D 02	0.0
301	0	NEWFL6P	10	5	0	112	0	0	0	0	47	44	0.29000000000000000000D 02	0.0
302	0	NEWFL6P	10	0	0	0	0	0	0	0	307	15	0.46862000000000000000D 02	0.0
303	0	NEWFL6P	10	0	0	0	0	0	0	0	0	0	0.52328000000000000000D 04	0.0
301	0	WNKFL6E	11	5	0	15	0	0	0	0	51	26	0.45583000000000000000D 02	0.0
302	0	WNKFL6E	11	0	0	0	0	0	0	0	359	18	0.79370000000000000000D 01	0.0
303	0	WNKFL6E	11	0	0	0	0	0	0	0	0	0	0.76068000000000000000D 02	0.0
301	0	WNKFL6P	12	5	0	115	0	0	0	0	51	26	0.45583000000000000000D 02	0.0
302	0	WNKFL6P	12	0	0	0	0	0	0	0	359	18	0.79070000000000000000D 01	0.0
303	0	WNKFL6P	12	0	0	0	0	0	0	0	0	0	0.76068000000000000000D 02	0.0
301	0	JOBUR6E	13	5	0	16	0	0	0	-1	25	53	0.27090000000000000000D 01	0.0
302	0	JOBUR6E	13	0	0	0	0	0	0	0	27	42	0.25420000000000000000D 02	0.0
303	0	JOBUR6E	13	0	0	0	0	0	0	0	0	0	0.15387760000000000000D 04	0.0
301	0	JOBUR6P	14	5	0	116	0	0	0	-1	25	53	0.27090000000000000000D 01	0.0
302	0	JOBUR6P	14	0	0	0	0	0	0	0	27	42	0.25420000000000000000D 02	0.0
303	0	JOBUR6P	14	0	0	0	0	0	0	0	0	0	0.15387760000000000000D 04	0.0
301	0	ULASK6E	15	5	0	19	0	0	0	0	64	58	0.35752000000000000000D 02	0.0
302	0	ULASK6E	15	0	0	0	0	0	0	0	212	28	0.30270000000000000000D 02	0.0
303	0	ULASK6E	15	0	0	0	0	0	0	0	0	0	0.26663400000000000000D 03	0.0
301	0	ULASK6P	16	5	0	119	0	0	0	0	64	58	0.30752000000000000000D 02	0.0
302	0	ULASK6P	16	0	0	0	0	0	0	0	212	28	0.30270000000000000000D 02	0.0
303	0	ULASK6P	16	0	0	0	0	0	0	0	0	0	0.26663400000000000000D 03	0.0
301	0	ORORA6E	17	5	0	21	0	0	0	-1	35	37	0.34095000000000000000D 02	0.0
302	0	ORORA6E	17	0	0	0	0	0	0	0	148	57	0.13588000000000000000D 02	0.0
303	0	ORORA6E	17	0	0	0	0	0	0	0	0	0	0.94015400000000000000D 03	0.0
301	0	ORORA6P	18	5	0	121	0	0	0	-1	35	37	0.34095000000000000000D 02	0.0
302	0	ORORA6P	18	0	0	0	0	0	0	0	148	57	0.13588000000000000000D 02	0.0
303	0	ORORA6P	18	0	0	0	0	0	0	0	0	0	0.94015400000000000000D 03	0.0
301	0	MAGDA6E	19	5	0	23	0	0	0	-1	19	0	0.33300000000000000000D 02	0.0
302	0	MAGDA6E	19	0	0	0	0	0	0	0	47	17	0.58900000000000000000D 02	0.0
303	0	MAGDA6E	19	0	0	0	0	0	0	0	0	0	0.13559000000000000000D 04	0.0
301	0	MAGDA6P	20	5	0	123	0	0	0	-1	19	0	0.33300000000000000000D 02	0.0
302	0	MAGDA6P	20	0	0	0	0	0	0	0	47	17	0.58900000000000000000D 02	0.0
303	0	MAGDA6P	20	0	0	0	0	0	0	0	0	0	0.13559000000000000000D 04	0.0

CONTROL DATA LISTING

CATEGORY	CONTINUE	LABEL	KEY1	KEY2	KEY3	KEY4	KEY5	KEY6	KEY7	KEY8	KEY9	KEY10	DATA1	DATA2
301	0	BPOINGE	21	5	0	1	0	0	0	0	38	25	0.49634000000000000000D 02	0.0
302	0	BPOINGE	21	0	0	0	0	0	0	0	262	54	0.48672000000000000000D 02	0.0
303	0	BPOINGE	21	0	0	0	0	0	0	0	0	0	-0.55908000000000000000D 02	0.0
301	0	BPOINGP	22	5	0	101	0	0	0	0	38	25	0.49634000000000000000D 02	0.0
302	0	BPOINGP	22	0	0	0	0	0	0	0	282	54	0.48672000000000000000D 02	0.0
303	0	BPOINGP	22	0	0	0	0	0	0	0	0	0	-0.55908000000000000000D 02	0.0
301	0	COLEGGE	23	5	0	13	0	0	0	0	64	52	0.17788000000000000000D 02	0.0
302	0	COLEGGE	23	0	0	0	0	0	0	0	212	9	0.36557000000000000000D 02	0.0
303	0	COLEGGE	23	0	0	0	0	0	0	0	0	0	0.13909000000000000000D 03	0.0
301	0	COLEGGP	24	5	0	113	0	0	0	0	64	52	0.17788000000000000000D 02	0.0
302	0	COLEGGP	24	0	0	0	0	0	0	0	212	9	0.36557000000000000000D 02	0.0
303	0	COLEGGP	24	0	0	0	0	0	0	0	0	0	0.13909000000000000000D 03	0.0
301	0	MOJAVGE	25	5	0	17	0	0	0	0	35	19	0.47630000000000000000D 02	0.0
302	0	MOJAVGE	25	0	0	0	0	0	0	0	243	5	0.59050000000000000000D 02	0.0
303	0	MOJAVGE	25	0	0	0	0	0	0	0	0	0	0.87272000000000000000D 03	0.0
301	0	MOJAVGP	26	5	0	117	0	0	0	0	35	19	0.47630000000000000000D 02	0.0
302	0	MOJAVGP	26	0	0	0	0	0	0	0	243	5	0.59050000000000000000D 02	0.0
303	0	MOJAVGP	26	0	0	0	0	0	0	0	0	0	0.87272000000000000000D 03	0.0
301	0	GFORKGE	27	5	0	14	0	0	0	0	43	1	0.20977000000000000000D 02	0.0
302	0	GFORKGE	27	0	0	0	0	0	0	0	262	59	0.19393000000000000000D 02	0.0
303	0	GFORKGE	27	0	0	0	0	0	0	0	0	0	0.19663400000000000000D 03	0.0
301	0	GFORKGP	28	5	0	114	0	0	0	0	45	1	0.20977000000000000000D 02	0.0
302	0	GFORKGP	28	0	0	0	0	0	0	0	262	59	0.19393000000000000000D 02	0.0
303	0	GFORKGP	28	0	0	0	0	0	0	0	0	0	0.19663400000000000000D 03	0.0
301	0	DOMERGE	29	5	0	16	0	0	0	-1	31	23	0.26964000000000000000D 02	0.0
302	0	DOMERGE	29	0	0	0	0	0	0	0	136	52	0.14226000000000000000D 02	0.0
303	0	DOMERGE	29	0	0	0	0	0	0	0	0	0	0.13000000000000000000D 03	0.0
301	0	DOMERGP	30	5	0	118	0	0	0	-1	31	23	0.26964000000000000000D 02	0.0
302	0	DOMERGP	30	0	0	0	0	0	0	0	136	52	0.14226000000000000000D 02	0.0
303	0	DOMERGP	30	0	0	0	0	0	0	0	0	0	0.13000000000000000000D 03	0.0
601	0	FTMYPELB	0	1	0	1	10	1	1	1	0	0	0.89817230931659090000D-04	0.10000000000000000D 03
601	0	FTMYPEMB	0	1	0	1	10	2	2	1	0	0	-0.12505939222038200000D-03	0.10000000000000000D 03
601	0	FTMYRPLB	0	2	0	1	10	3	3	1	0	0	0.10338503265401400000D-03	0.10000000000000000D 03
601	0	FTMYRPMB	0	2	0	1	10	4	4	1	0	0	-0.50214156779718600000D-03	0.10000000000000000D 03
601	0	GUITOELB	0	3	0	1	10	5	5	1	0	0	0.67043673526717490000D-03	0.10000000000000000D 03
601	0	GUITOEMB	0	3	0	1	10	6	6	1	0	0	0.11872657099083700000D-02	0.10000000000000000D 03
601	0	GUITOPLB	0	4	0	1	10	7	7	1	0	0	0.14436949250698300000D-03	0.10000000000000000D 03
601	0	GUITOPMB	0	4	0	1	10	8	8	1	0	0	-0.31296226369007190000D-03	0.10000000000000000D 03
601	0	LIMAPELB	0	5	0	1	10	9	9	1	0	0	0.0	0.10000000000000000D 03
601	0	LIMAPEMB	0	5	0	1	10	10	10	1	0	0	0.0	0.10000000000000000D 03
601	0	LIMAPPLB	0	6	0	1	10	11	11	1	0	0	0.30906913777113600000D-03	0.10000000000000000D 03
601	0	LIMAPPMB	0	6	0	1	10	12	12	1	0	0	-0.88750460516658390000D-04	0.10000000000000000D 03
601	0	SNTAGELB	0	7	0	1	10	13	13	1	0	0	-0.32619604117238600000D-04	0.10000000000000000D 03
601	0	SNTAGEMB	0	7	0	1	10	14	14	1	0	0	0.40725431369300490000D-03	0.10000000000000000D 03
601	0	SNTAGPLB	0	8	0	1	10	15	15	1	0	0	0.92011013546497290000D-04	0.10000000000000000D 03
601	0	SNTAGPMB	0	8	0	1	10	16	16	1	0	0	0.31154163075425500000D-03	0.10000000000000000D 03
601	0	NEWFLELB	0	9	0	1	10	17	17	1	0	0	0.13328548974420500000D-03	0.10000000000000000D 03
601	0	NEWFLEMB	0	9	0	1	10	18	18	1	0	0	-0.12217775290375000000D-02	0.10000000000000000D 03
601	0	NEWFLPLB	0	10	0	1	10	19	19	1	0	0	0.55097828484590690000D-04	0.10000000000000000D 03
601	0	NEWFLPMB	0	10	0	1	10	20	20	1	0	0	-0.53496975105880390000D-03	0.10000000000000000D 03

CONTROL DATA LISTING

CATEGORY	CONTINUE	LABEL	KEY1	KEY2	KEY3	KEY4	KEY5	KEY6	KEY7	KEY8	KEY9	KEY10	DATA1	DATA2
601	0	WNKFLELB	0	11	0	1	10	21	21	1	0	0	-0.102613000752380000000D-04	0.10000000000000000D 03
601	0	WNKFLEMB	0	11	0	1	10	22	22	1	0	0	-0.67330434562159450000D-03	0.10000000000000000D 03
601	0	WNKFLPLB	0	12	0	1	10	23	23	1	0	0	0.33558145008001000000D-04	0.10000000000000000D 03
601	0	WNKFLPMB	0	12	0	1	10	24	24	1	0	0	-0.32405210058711800000D-03	0.10000000000000000D 03
601	0	JOBURELB	0	13	0	1	10	25	25	1	0	0	-0.11232775829064700000D-03	0.10000000000000000D 03
601	0	JOBUREMB	0	13	0	1	10	26	26	1	0	0	0.22610553317395700000D-03	0.10000000000000000D 03
601	0	JOBURPLB	0	14	0	1	10	27	27	1	0	0	-0.47222130573455500000D-04	0.10000000000000000D 03
601	0	JOBURPMB	0	14	0	1	10	28	28	1	0	0	-0.23467027486256700000D-03	0.10000000000000000D 03
601	0	ULASKELB	0	15	0	1	10	29	29	1	0	0	-0.42107941655363300000D-05	0.10000000000000000D 03
601	0	ULASKEMB	0	15	0	1	10	30	30	1	0	0	-0.24702572795517645000D-03	0.10000000000000000D 03
601	0	ULASKPLB	0	16	0	1	10	31	31	1	0	0	-0.17592250643751100000D-03	0.10000000000000000D 03
601	0	ULASKPMB	0	16	0	1	10	32	32	1	0	0	-0.56696035243763690000D-03	0.10000000000000000D 03
601	0	PRORAELB	0	17	0	1	10	33	33	1	0	0	0.73476711731964850000D-04	0.10000000000000000D 03
601	0	PRORAEMB	0	17	0	1	10	34	34	1	0	0	0.22409617581553890000D-04	0.10000000000000000D 03
601	0	PRORAPLB	0	18	0	1	10	35	35	1	0	0	0.11712625397798200000D-03	0.10000000000000000D 03
601	0	PRORAPMB	0	18	0	1	10	36	36	1	0	0	-0.37656623274501900000D-03	0.10000000000000000D 03
601	0	MAGDAELB	0	19	0	1	10	37	37	1	0	0	-0.74383948528624000000D-04	0.10000000000000000D 03
601	0	MAGDAEMB	0	19	0	1	10	38	38	1	0	0	-0.59330835224193490000D-03	0.10000000000000000D 03
601	0	MAGDAPLB	0	20	0	1	10	39	39	1	0	0	0.10754011875318300000D-04	0.10000000000000000D 03
601	0	MAGDAPMB	0	20	0	1	10	40	40	1	0	0	-0.28974794621531800000D-03	0.10000000000000000D 03
601	0	FTMYREMS	0	1	0	1	14	2	41	1	0	0	0.99413065932971600000D-14	0.10000000000000000D-02
601	0	FTMYRPLS	0	2	0	1	14	3	42	1	0	0	-0.68337096355621890000D-03	0.10000000000000000D-02
601	0	GUITOEMS	0	3	0	1	14	6	43	1	0	0	-0.10144573785853600000D-12	0.10000000000000000D-02
601	0	GUITOPLS	0	4	0	1	14	7	44	1	0	0	-0.83444727867006590000D-03	0.10000000000000000D-02
601	0	LIMAPEMS	0	5	0	1	14	10	45	1	0	0	0.0	0.10000000000000000D-02
601	0	LIMAPPLS	0	6	0	1	14	11	46	1	0	0	-0.13285772114190000000D-03	0.10000000000000000D-02
601	0	SNTAGEMS	0	7	0	1	14	14	47	1	0	0	-0.10535916895697900000D-03	0.10000000000000000D-02
601	0	SNTAGPLS	0	8	0	1	14	15	48	1	0	0	-0.75530162726271590000D-03	0.10000000000000000D-02
601	0	NEWFLEMS	0	9	0	1	14	18	49	1	0	0	-0.21083490608141200000D-03	0.10000000000000000D-02
601	0	NEWFLPLS	0	10	0	1	14	19	50	1	0	0	-0.49733464437246690000D-03	0.10000000000000000D-02
601	0	WNKFLEMS	0	11	0	1	14	22	51	1	0	0	0.35868451106368200000D-04	0.10000000000000000D-02
601	0	WNKFLPLS	0	12	0	1	14	23	52	1	0	0	-0.25274761722941790000D-03	0.10000000000000000D-02
601	0	JOBUREMS	0	13	0	1	14	26	53	1	0	0	-0.10414410659067500000D-05	0.10000000000000000D-02
601	0	JOBURPLS	0	14	0	1	14	27	54	1	0	0	-0.46791736844572390000D-03	0.10000000000000000D-02
601	0	ULASKEMS	0	15	0	1	14	30	55	1	0	0	-0.42448633526959590000D-03	0.10000000000000000D-02
601	0	ULASKPLS	0	16	0	1	14	31	56	1	0	0	-0.25457194513191900000D-03	0.10000000000000000D-02
601	0	PRORAEMS	0	17	0	1	14	34	57	1	0	0	-0.19295550355702900000D-04	0.10000000000000000D-02
601	0	PRORAPLS	0	18	0	1	14	35	58	1	0	0	-0.29845032244408000000D-03	0.10000000000000000D-02
601	0	MAGDAEMS	0	19	0	1	14	38	59	1	0	0	-0.72808836254579000000D-04	0.10000000000000000D-02
601	0	MAGDAPLS	0	20	0	1	14	39	60	1	0	0	-0.23318078387040200000D-03	0.10000000000000000D-02
601	0	FTMYRELR	0	1	0	1	15	1	61	1	0	0	0.15844699128206100000D-04	0.10000000000000000D-02
601	0	FTMYRPMR	0	2	0	1	15	4	62	1	0	0	-0.75946654648661990000D-04	0.10000000000000000D-02
601	0	GUITOELR	0	3	0	1	15	5	63	1	0	0	-0.57290723808134890000D-13	0.10000000000000000D-02
601	0	GUITOPMR	0	4	0	1	15	8	64	1	0	0	0.13864045002216500000D-04	0.10000000000000000D-02
601	0	LIMAPELR	0	5	0	1	15	9	65	1	0	0	0.0	0.10000000000000000D-02
601	0	LIMAPPMR	0	6	0	1	15	12	66	1	0	0	0.89709648046666490000D-05	0.10000000000000000D-02
601	0	SNTAGELR	0	7	0	1	15	13	67	1	0	0	0.29458286206173590000D-03	0.10000000000000000D-02
601	0	SNTAGPMR	0	8	0	1	15	16	68	1	0	0	0.10052146552980800000D-02	0.10000000000000000D-02
601	0	NEWFLELR	0	9	0	1	15	17	69	1	0	0	0.15483602756615790000D-04	0.10000000000000000D-02
601	0	NEWFLPMR	0	10	0	1	15	20	70	1	0	0	0.20724828825630700000D-03	0.10000000000000000D-02

CONTROL DATA LISTING

CATEGORY	CONTINUE	LABEL	KEY1	KEY2	KEY3	KEY4	KEY5	KEY6	KEY7	KEY8	KEY9	KEY10	DATA1	DATA2
601	0	WNKFLELR	0	11	0	1	15	21	71	1	0	0	-0.43205661581860500000D-04	0.1000000000000000D-02
601	0	WNKFLPMR	0	12	0	1	15	24	72	1	0	0	0.13657565430198300000D-03	0.1000000000000000D-02
601	0	JOBURELR	0	13	0	1	15	25	73	1	0	0	-0.65001370117290900000D-06	0.1000000000000000D-02
601	0	JOBURPMR	0	14	0	1	15	28	74	1	0	0	0.15454926146284600000D-04	0.1000000000000000D-02
601	0	ULASKELR	0	15	0	1	15	29	75	1	0	0	0.73421271203233600000D-05	0.1000000000000000D-02
601	0	ULASKPMR	0	16	0	1	15	32	76	1	0	0	0.42906404819012960000D-04	0.1000000000000000D-02
601	0	ORDRAELR	0	17	0	1	15	33	77	1	0	0	0.10094545999512500000D-03	0.1000000000000000D-02
601	0	ORDRAPMR	0	18	0	1	15	36	78	1	0	0	0.13009161757038800000D-03	0.1000000000000000D-02
601	0	MAGDAELR	0	19	0	1	15	37	79	1	0	0	0.10932285030558400000D-03	0.1000000000000000D-02
601	0	MAGDAPMR	0	20	0	1	15	40	80	1	0	0	0.21994475581006200000D-03	0.1000000000000000D-02
601	0	FTMYER T	0	0	0	1	11	1	81	1	0	0	-0.18405766881678290000D-02	0.1000000000000000D-01
602	0	FTMYER T	11	1	2	3	4	0	0	0	0	0	0.0	0.0
601	0	QUITO T	0	0	0	1	11	5	82	1	0	0	-0.53887599270076900000D-04	0.1000000000000000D-01
602	0	QUITO T	11	5	5	7	8	0	0	0	0	0	0.0	0.0
601	0	LIMA T	0	0	0	1	11	9	83	1	0	0	0.63840253125036000000D-04	0.1000000000000000D-01
602	0	LIMA T	11	9	10	11	12	0	0	0	0	0	0.0	0.0
601	0	SANTIA T	0	0	0	1	11	13	84	1	0	0	0.44654800070837990000D-02	0.1000000000000000D-01
602	0	SANTIA T	11	13	14	15	16	0	0	0	0	0	0.0	0.0
601	0	NEWFLD T	0	0	0	1	11	17	85	1	0	0	-0.15800831104863700000D-02	0.1000000000000000D-01
602	0	NEWFLD T	11	17	18	19	20	0	0	0	0	0	0.0	0.0
601	0	WNKFLD T	0	0	0	1	11	21	86	1	0	0	0.12020727410470900000D-02	0.1000000000000000D-01
602	0	WNKFLD T	11	21	22	23	24	0	0	0	0	0	0.0	0.0
601	0	JOBURG T	0	0	0	1	11	25	87	1	0	0	-0.10344628130102200000D-02	0.1000000000000000D-01
602	0	JOBURG T	11	25	26	27	28	0	0	0	0	0	0.0	0.0
601	0	ULASKA T	0	0	0	1	11	29	88	1	0	0	-0.70296693003234690000D-04	0.1000000000000000D-01
602	0	ULASKA T	11	29	30	31	32	0	0	0	0	0	0.0	0.0
601	0	ORDRAL T	0	0	0	1	11	33	89	1	0	0	-0.14570386573839500000D-02	0.1000000000000000D-01
602	0	ORDRAL T	11	33	34	35	36	0	0	0	0	0	0.0	0.0
601	0	MADGAR T	0	0	0	1	11	37	90	1	0	0	0.31247150703872600000D-03	0.1000000000000000D-01
602	0	MADGAR T	11	37	38	39	40	0	0	0	0	0	0.0	0.0
205	0		1	0	0	0	0	68	2	27	23	56	0.58400000000000000000D 02	0.0
601	0	X	1	0	0	2	1	0	1	0	0	0	0.21213259590637300000D 07	0.1000000000000000D 10
601	0	Y	1	0	0	2	2	0	2	0	0	0	0.78072302096756950000D 05	0.1000000000000000D 10
601	0	Z	1	0	0	2	3	0	3	0	0	0	0.71460581726226950000D 07	0.1000000000000000D 10
601	0	XO	1	0	0	2	4	0	4	0	0	0	0.23477872805223900000D 04	0.1000000000000000D 10
601	0	YO	1	0	0	2	5	0	5	0	0	0	-0.70192657106413800000D 04	0.1000000000000000D 10
601	0	ZO	1	0	0	2	6	0	6	0	0	0	-0.60802815336164300000D 03	0.1000000000000000D 10
701	0	FTMEOL	1	1	6	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	FTMEOM	2	1	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	FTMPOL	3	2	6	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	FTMPOM	4	2	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	OTOEOL	5	3	6	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	OTOEOM	6	3	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	OTOPOL	7	4	6	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	OTOPOM	8	4	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	LMAEOL	9	5	6	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	LMAEOM	10	5	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	LMAPOL	11	6	6	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	LMAPOM	12	6	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	STAEOL	13	7	6	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03

CONTROL DATA LISTING

CATEGORY	CONTINUE	LABEL	KEY1	KEY2	KEY3	KEY4	KEY5	KEY6	KEY7	KEY8	KEY9	KEY10	DATA1	DATA2
701	0	STAEQM	14	7	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	STAPOL	15	8	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	STAPOM	16	8	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	NFLEQL	17	9	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	NFLEQM	18	9	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	NFLPOL	19	10	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	NFLPOM	20	10	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	WKFEQL	21	11	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	WKFEQM	22	11	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	WKFPOL	23	12	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	WKFPOM	24	12	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	JBREQL	25	13	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	JBREQM	26	13	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	JBRPOL	27	14	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	JBRPOM	28	14	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	LSKEQL	29	15	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	LSKEQM	30	15	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	LSKPOL	31	16	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	LSKPOM	32	16	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	ORAEQL	33	17	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	ORAEQM	34	17	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	ORAPOL	35	18	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	ORAPOM	36	18	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	MGAEQL	37	19	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	MGAEQM	38	19	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	MGAPOL	39	20	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	MGAPOM	40	20	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	DPNEQL	41	21	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	DPNEQM	42	21	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	DPNPOL	43	22	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	DPNPOM	44	22	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	CLGEQL	45	23	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	CLGEQM	46	23	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	CLGPOL	47	24	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	CLGPOM	48	24	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	MJVEQL	49	25	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	MJVEQM	50	25	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	MJVPOL	51	26	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	MJVPOM	52	26	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	GFKEQL	53	27	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	GFKEQM	54	27	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	GFKPOL	55	28	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	GFKPOM	56	28	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	CMREQL	57	29	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	CMREQM	58	29	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	CMRPOL	59	30	5	3	0	0	0	0	0	0	0.0	0.3000000000000000D-03
701	0	CMRPOM	60	30	7	4	0	0	0	0	0	0	0.0	0.3000000000000000D-03
702	0	FTMEQL	1	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	FTMEQM	2	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	FTMPOL	3	0	1	0	0	0	0	0	0	0	0.0	0.0

CONTROL DATA LISTING

CATEGORY	CONTINUE	LABEL	KEY1	KEY2	KEY3	KEY4	KEY5	KEY6	KEY7	KEY8	KEY9	KEY10	DATA1	DATA2
702	0	FTMPOM	4	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GTOEOL	5	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GTOEOM	6	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GTOPOL	7	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GTOPOM	8	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	LMAEOL	9	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	LMAEOM	10	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	LMAPOL	11	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	LMAPOM	12	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	STAEOL	13	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	STAEOM	14	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	STAPOL	15	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	STAPOM	16	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	NFLEOL	17	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	NFLEOM	18	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	NFLPOL	19	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	NFLPOM	20	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	WKFEOL	21	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	WKFEOM	22	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	WKFPOL	23	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	WKFPOM	24	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	JBREOL	25	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	JBREOM	26	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	JBRPOL	27	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	JBRPOM	28	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	LSKEOL	29	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	LSKEOM	30	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	LSKPOL	31	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	LSKPOM	32	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GRAEOL	33	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GRAEOM	34	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GRAPOL	35	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GRAPOM	36	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	MGAEOL	37	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	MGAEOM	38	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	MGAPOL	39	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	MGAPOM	40	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GPNEOL	41	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GPNEOM	42	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GPNPOL	43	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GPNPOM	44	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	CLGEOL	45	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	CLGEOM	46	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	CLGPOL	47	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	CLGPOM	48	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	MJVEOL	49	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	MJVEOM	50	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	MJVPOL	51	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	MJVPOM	52	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GFKEOL	53	0	1	0	0	0	0	0	0	0	0.0	0.0

CONTROL DATA LISTING

CATEGORY	CONTINUE	LABEL	KEY1	KEY2	KEY3	KEY4	KEY5	KEY6	KEY7	KEY8	KEY9	KEY10	DATA1	DATA2
702	0	GFKEQM	54	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GFKPOL	55	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GFKPOM	56	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GMREOL	57	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	GMREOM	58	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	CMRPOL	59	0	1	0	0	0	0	0	0	0	0.0	0.0
702	0	CMRPOM	60	0	1	0	0	0	0	0	0	0	0.0	0.0
201	0	LIMAP6P	1	6	1	0	0	68	1	58	0	0	0.34370000000000000000D 04	0.0
202	0	LIMAP6P	1	6	1	0	0	68	1	58	0	0	0.34678000000000000000D 04	0.0
701	1	LMAPOL	11	6	5	3	0	0	0	0	0	0	0.10000000000000000000D 01	0.23415645000000000D-02
701	1	LMAPOM	12	6	7	4	0	0	0	0	0	0	0.10000000000000000000D 01	0.16245144000000000D-02
999	0	LIMAP6P	1	6	1	1	0	0	0	0	0	0	0.0	0.0
201	0	GUITOCP	1	4	1	0	0	66	1	58	0	0	0.36390000000000000000D 04	0.0
202	0	GUIT06P	1	4	1	0	0	68	1	58	0	0	0.35918000000000000000D 04	0.0
701	1	GTOPOL	7	4	5	3	0	0	0	0	0	0	0.10000000000000000000D 01	0.30817056000000000D-02
701	1	GTOPOM	8	4	7	4	0	0	1	0	0	0	0.10000000000000000000D 01	0.26676641000000000D 04
999	0	GUIT06P	1	4	1	1	0	0	0	0	0	0	0.0	0.0
201	0	FTMYR6P	1	2	1	0	0	68	1	58	0	0	0.41510000000000000000D 04	0.0
202	0	FTMYR6P	1	2	1	0	0	68	1	58	0	0	0.42137999999999990000D 04	0.0
701	1	FTMPOL	3	2	5	3	0	0	0	0	0	0	0.10000000000000000000D 01	0.64512718000000000D-03
701	1	FTMPOM	4	2	7	4	0	0	0	0	0	0	0.10000000000000000000D 01	0.60432183000000000D-03
999	0	FTMYR6P	1	2	1	1	0	0	0	0	0	0	0.0	0.0
201	0	FTMYR6E	1	1	1	0	0	58	1	58	0	0	0.44830000000000000000D 04	0.0
202	0	FTMYR6E	1	1	1	0	0	68	1	58	0	0	0.43457999999999990000D 04	0.0
701	1	FTMEOL	1	1	5	3	0	0	0	0	0	0	0.10000000000000000000D 01	0.46342446000000000D-02
701	1	FTMEOM	2	1	7	4	0	0	1	0	0	0	0.10000000000000000000D 01	0.41260638000000000D 04
999	0	FTMYR6E	1	1	1	1	0	0	0	0	0	0	0.0	0.0
201	0	ULASK6E	1	15	1	0	0	68	1	58	0	0	0.52090000000000000000D 04	0.0
202	0	ULASK6E	1	15	1	0	0	68	1	58	0	0	0.52717999999999990000D 04	0.0
701	1	LSKEOL	29	15	5	3	0	0	1	0	0	0	0.10000000000000000000D 01	0.10203662000000000D 05
701	1	LSKEOM	30	15	7	4	0	0	0	0	0	0	0.10000000000000000000D 01	0.10327337000000000D-02
999	0	ULASK6E	1	15	1	1	0	0	0	0	0	0	0.0	0.0
201	0	SNTAG6P	1	8	1	0	0	68	1	58	0	0	0.97290000000000000000D 04	0.0
202	0	SNTAG6P	1	8	1	0	0	68	1	58	0	0	0.97918000000000000000D 04	0.0
701	1	STAPOL	15	8	5	3	0	0	0	0	0	0	0.10000000000000000000D 01	0.57155631000000000D-03
701	1	STAPOM	16	8	7	4	0	0	0	0	0	0	0.10000000000000000000D 01	0.55220759000000000D-03
999	0	SNTAG6P	1	8	1	1	0	0	0	0	0	0	0.0	0.0
201	0	LIMAP6P	1	6	2	0	0	68	1	58	0	0	0.10157000000000000000D 05	0.0
202	0	LIMAP6P	1	6	2	0	0	68	1	58	0	0	0.10219800000000000000D 05	0.0
701	1	LMAPOL	11	6	5	3	0	0	1	0	0	0	0.10000000000000000000D 01	0.16271639000000000D 05
701	1	LMAPOM	12	6	7	4	0	0	0	0	0	0	0.10000000000000000000D 01	0.14624656000000000D-02
999	0	LIMAP6P	1	6	2	1	0	0	0	0	0	0	0.0	0.0
201	0	FTMYR6P	1	2	2	0	0	68	1	58	0	0	0.10905000000000000000D 05	0.0
202	0	FTMYR6P	1	2	2	0	0	68	1	58	0	0	0.10955180000000000000D 05	0.0
704	1	FTMPOM	4	0	0	0	0	0	0	0	0	0	-0.17502676879993400000D-01	0.0
701	1	FTMPOL	3	2	5	3	0	0	1	0	0	0	0.10000000000000000000D 01	0.43829621000000000D 04
701	1	FTMPOM	4	2	7	4	0	0	0	0	0	0	0.10000000000000000000D 01	0.41466811000000000D-02
999	0	FTMYR6P	1	2	2	1	0	0	0	0	0	0	0.0	0.0
201	0	ULASK6E	1	15	2	0	0	68	1	58	0	0	0.11783000000000000000D 05	0.0
202	0	ULASK6E	1	15	2	0	0	68	1	58	0	0	0.11845800000000000000D 05	0.0

2.7 REFERENCES

1. Watkins, Jr., Edward R., "Preprocessing of Minitrack Data", Goddard Space Flight Center, May 1969, NASA TN-D-5042.

2. Control Systems Research, Inc., "Minitrack Tracking Function Description", March 1970, Final Report NASA Contract NAS5-10694.

3. Rice, William M., "A Review of NASA Minitrack Data Time-Tagging", November 1970, Goddard Space Flight Center X-551-70-41.

4. DBA Systems, Inc., "Network Analysis Program, Phase II, User's Guide to Data Preparation", December 1969, NASA Contract NAS5-10588.

5. Morduch, G. E., "An Algorithm for the Computation of the Gradient Of a Newtonian Potential Expressed as a Sum of Spherical Harmonics", June 1971, NASA/GSFC X-551-71-248.

APPENDIX A-1

CONSTANTS USED IN NAP-II

A-1.1 THE GRAVITY MODEL USED

The gravity model used for data processing under this contract was the Smithsonian M1 model, (Reference 1), which is built into the NAP program, modified by the GEOS-A resonant harmonics (Reference 2). The GEOS-A resonant harmonics were not obtained from the original source, but from (Reference 3).

The gravitational mass of the Earth was taken as .3986032 D15 m^3/sec^2, which is the NAP default value.

The Smithsonian M1 values for the spherical harmonics coefficients are listed below:

(N, M)	C(N,M)	S(N,M)
(0, 0)	1.	0.
(1, 0)	0.	0.
(2, 0)	- .108264500002 D-2	0.
(3, 0)	.254599999999 D-5	0.
(4, 0)	.164900000002 D-5	0.
(5, 0)	.21 D-6	0.
(6, 0)	- .645999999993 D-6	0.
(7, 0)	.332999999998 D-6	0.
(8, 0)	.269999999997 D-6	0.
(9, 0)	.529999999998 D-7	0.
(10, 0)	.540000000002 D-7	0.
(11, 0)	- .302 D-6	0.
(12, 0)	.357000000006 D-6	0.
(13, 0)	.113999999999 D-6	0.
(14, 0)	- .178999999999 D-6	0.
(1, 1)	0.	0.
(2, 1)	0.	0.
(3, 1)	.209111899862 D-5	.287312837632 D-6
(4, 1)	- .542646846485 D-6	- .444932466779 D-6
(5, 1)	- .676515582477 D-7	- .88203930748 D-7
(6, 1)	- .36979402246 D-7	- .21243486397 D-7
(7, 1)	.14418923578 D-6	.114180308536 D-6
(8, 1)	- .515388203208 D-7	.446669776114 D-7
(9, 1)	.76024995806 D-7	.779743547594 D-8
(10, 1)	.648810099708 D-7	- .778572119639 D-7
(11, 1)	- .312872807930 D-7	.885489079044 D-8
(12, 1)	- .922805782847 D-7	- .401958347128 D-7
(13, 1)	0.	0.
(14, 1)	- .788307409236 D-8	.278535284599 D-8

(N,M)	C(N,M)		S(N,M)	
(2, 2)	.153563789676	D-5	- .872066750118	D-6
(3, 2)	.250708728736	D-6	- .183761983737	D-6
(4, 2)	.737902432577	D-7	.147804093315	D-6
(5, 2)	.102117707526	D-6	- .375456547484	D-7
(6, 2)	.858383031385	D-8	- .455316216648	D-7
(7, 2)	.362552678174	D-7	.162351886106	D-7
(8, 2)	.21354899813	D-8	.320323497197	D-8
(9, 2)	- .27706980487	D-9	.242436079257	D-8
(10, 2)	- .624317809516	D-8	- .249727123811	D-8
(11, 2)	0.		0.	
(12, 2)	- .469893640425	D-8	- .232665783121	D-9
(3, 3)	.782277124813	D-7	.225898207164	D-6
(4, 3)	.508569773273	D-7	- .113546717887	D-7
(5, 3)	- .171778742888	D-7	.231240615425	D-9
(6, 3)	- .111963004094	D-8	.642750579065	D-9
(7, 3)	.352147606138	D-8	.253546276422	D-9
(8, 3)	- .374070918995	D-9	.404400993503	D-10
(9, 3)	0.		0.	
(10, 3)	- .378977225408	D-9	.174912565571	D-9
(4, 4)	- .11198293875	D-8	.485963696475	D-8
(5, 4)	- .206336328204	D-8	.498321698304	D-9
(6, 4)	- .166560812946	D-9	- .196087502512	D-8
(7, 4)	- .322776554575	D-9	- .216600056362	D-9
(8, 4)	- .276702101966	D-9	- .156623831302	D-10
(9, 4)	0.		0.	
(10, 4)	- .435831320535	D-10	- .653746980798	D-10
(5, 5)	.384108946188	D-9	- .145764420605	D-8
(6, 5)	- .252611520257	D-9	- .369636026451	D-9
(7, 5)	.268980462153	D-10	.191117696793	D-10
(8, 5)	- .959291939079	D-11	.213578205308	D-10
(6, 6)	- .931919036559	D-11	- .36111862666	D-10
(7, 6)	- .145066373826	D-10	.437281413924	D-11
(8, 6)	- .474787418211	D-12	.88813175875	D-11

(N,M)	C(N,M)	S(N,M)
(7, 7)	.102027945339 D-11	.178085140954 D-11
(8, 7)	- .443617740607 D-13	.158070689181 D-12
(8, 8)	- .316141378363 D-12	.130025889487 D-12
(12, 12)	0.	0.
(13, 12)	- .1082186306 D-18	.895602460152 D-19
(14, 12)	0.	0.
(15, 12)	- .114596608272 D-19	.107203278705 D-19
(13, 13)	- .274440770475 D-19	.365921027295 D-20
(14, 13)	0.	0.
(15, 13)	- .127051846403 D-20	- .133101934329 D-20
(14, 14)	0.	0.
(15, 14)	.219788278315 D-22	- .532258360739 D-22

The GEOS-A resonant spherical harmonics coefficients are listed below:

(N,M)	C(N,M)	S(N,M)
(13,12)	-.126299 D-18	.165220 D-18
(14,12)	.139978 D-20	-.131772 D-19
(15,12)	-.138126 D-19	-.189639 D-20

REFERENCES

1. Lundquist, C. A., Veis, G., "Geodetic Parameters for a 1966 Smithsonian Institution Standard Earth," SAO Special Report Number 200, Volume 1.

2. Kohnlein, W., "The Earth's Gravitational Field as Derived From a Combination of Satellite Data With Gravity Anomalies," Prepared for XIV General Assembly, International Union of Geodesy and Geophysics, International Association of Geodesy, October 1967.

3. Lerch, F. J., Marsh, J. G., O'Neill, B., "Evaluation of the Goddard Range and Range Rate System at Rosman by Intercomparison With GEOS-I Long Arc Orbital Solutions," Goddard Space Flight Center, Maryland, X-552-68-72, November 1967.

A-1.2 THE EARTH MODEL USED AND STATION COORDINATES

The C-5 Earth Model was adopted for the data processing. According to this model the Earth is defined geometrically by the following constants:

Rotation Rate	=	.7292115854937 D-4 radians/sec
Semi-Major Axis	=	6378165 meters
Eccentricity Squared	=	.669454185459 D-2.

The station coordinates are given below:

Station Coordinates (Optical)

Station	Station ID	Latitude	Longitude	Height Above Geoid
COLDLK	29424	54°44'37.26"	249°57'21.90"	548 meters
NEWFL	31032	47°44'28.73"	307°16'46.67"	58 meters
JUPTH	37073	27°1'14.33"	279°53'12.72"	- 41 meters
JUPTR	29010	27°1'14.23"	279°53'12.95"	- 36 meters
JBC4L	37074	27°1'14.55"	279°53'12.76"	- 38 meters
COLEG	31033	64°52'17.78"	212°9'37.29"	139 meters
OOMER	31024	-31°23'26.96"	136°52'14.25"	148 meters
EDINB	37036	26°22'46.35"	261°40'7.34"	15 meters
JUP40	37072	27°1'14.39"	279°53'12.49"	- 38 meters
GFORK	31034	48°1'20.81"	262°59'19.55"	200 meters
ROSMA	31042	35°12'7.03"	277°7'40.81"	857 meters
PURIO	37040	18°15'28.30"	294°0'23.63"	5 meters
ORGAN	29001	32°25'24.70"	253°26'48.29"	1610 meters
GSFCP	37043	39°1'14.78"	283°10'20.39"	- 1 meter
BPOIN	31021	38°25'49.44"	282°54'48.65"	- 50 meters
EDWAFB	29425	34°57'50.17"	242°5'7.80"	754 meters

Station Coordinates (Optical) (Continued)

Station	Station ID	Latitude	Longitude	Height Above Geoid
JUP24	37071	27°1'14.00"	279°53'12.30"	- 38 meters
DENVR	37045	39°38'47.54"	255°23'38.52"	1751 meters
COLBA	37037	38°53'35.81"	267°47'40.85"	218 meters
FTMYR	31022	26°32'53.08"	278°8'3.80"	- 42 meters
BERMD	37039	32°21'48.94"	295°20'34.18"	- 28 meters
MOJAV	31030	35°19'47.57"	243°5'59.18"	874 meters
SATAG	31028	-33°8'58.76"	289°19'52.59"	705 meters

Station Coordinates (Minitrack)

Station	Latitude	Longitude	Height Above Geoid
Fort Myers	26°32'53.08"	278°8'3.80"	- 42 meters
Quito	-37'22.63"	281°25'15.23"	3554 meters
Lima	-11°46'37.56"	282°50'58.86"	34 meters
Santiago	-33°8'58.76"	289°19'52.59"	705 meters
New Foundland	47°44'28.73"	307°16'46.67"	58 meters
Winkfield	51°26'40.67"	359°18'8.35"	76 meters
Johannesburg	-25°53'2.7"	27°42'25.41"	1546 meters
Blossom Point	38°25'49.44"	282°54'48.65"	- 50 meters
College	64°52'17.78"	212°9'37.29"	139 meters
Mojave	35°19'47.57"	243°5'59.18"	874 meters
Grand Fork	48°1'20.81"	262°59'19.55"	200 meters
Woomera	-31°23'26.96"	136°52'14.25"	148 meters

APPENDIX A-2

SPECIAL PROGRAMS WRITTEN TO AID MINITRACK DATA REDUCTION

A-2.1 PROGRAM FOR EXTRACTING MINITRACK MESSAGES FOR GEOS I AND SAMPLE JCL

```
C
C     PROGRAM FOR EXTRACTING MINITRACK MESSAGES FOR SATELLITE ID 65891
C
C
       DIMENSION D(65),T(8)
       DATA T/'&','6','5','8','9','1',' ',' '/
       LOGICAL*1 D,T,D1,T1
       REAL*8  DATA,TEST
       EQUIVALENCE (DATA,D1,D(1)),(TEST,T1,T(1))
      REWIND  9
      REWIND 11
       NPASST = 0
       NPASSG = 0
  100 READ (9, 601, END=900, ERR=100)     D
       IF(D1.NE.T1) GO TO 100
  200  NPASST = NPASST + 1
        IF (DATA.NE.TEST) GO TO 100
      NPASSG=NPASSG+    1
       WRITE(6,602) NPASST, NPASSG
  300 WRITE(11,601) D
  400  READ(9,601,END=900,ERR=400) D
       IF(D1.EQ.T1) GO TO 200
       GO TO 300
  900  CONTINUE
      REWIND  9
       END FILE 11
      REWIND 11
       DO 940 I =1,100
  910  CONTINUE
       READ(11,601,END=950,ERR=910)  D
       WRITE(6,603)    D
  940  CONTINUE
  950  CONTINUE
       REWIND 11
      STOP
  601 FORMAT (65A1)
  602 FORMAT (//2X,'TOTAL NUMBER OF MESSAGES', I10/
     *           2X,'NUMBER OF GOES-A MESSAGES',I10)
  603  FORMAT (10X,65A1)
        STOP
        END
```

```
/*
//STEP2 EXEC LINKGO
//GO.FT09F001 DD UNIT=(9TRACK,,DEFER),DISP=(OLD,KEEP),LABEL=(2,BLP),
// DCB=(RECFM=FB,BLKSIZE=3250,LRECL=65,DEN=2),VOL=SER=31027G,DSN=&DEO
//GO.FT11F001 DD UNIT=2400-9,LABEL=(7,BLP),DISP=(NEW,DELETE),
// DCB=(RECFM=FB,BLKSIZE=3250,LRECL=65,DEN=2),VOL=SER=33951C,DSN=&MOR
/*
```

0059 CARDS

A-2.2 MINITRACK SORT PROGRAM AND JCL

```
//Z7GEMSRT JOB (G70041150A,T,C00138,007007),Z7,MSGLEVEL=1
//STEP1 EXEC FORTRANH,PARM='OPT=2'
//SOURCE.SYSIN DD *
      DEFINE FILE 11(120C,2145,L,ID11)
      DIMENSION BLANKS(65),AMPIC(65),MINI(65,33),DATA(65),MINY(2145)
      DIMENSION WORD(1202),LRECD(1200),K1(8),K2(8),K3(8),KD(8)
      DIMENSION KSTA(16),STATIC(16),JCNE(10),MON(14)
      LOGICAL*1 BLANKS,AMPID,MINI,DATA,MINY,BIN,PEZ,D1,AMP
      REAL*8 STATIO
      REAL*8 INTAPE,NDTAPE
      INTEGER*2 K1,K2,K3,KD,KSTA,NSTAD,NSTAT,L,LINED
      CCMPLEX*16 WORD,WORC1,WCRD2,WORD3,WORDD
      EQUIVALENCE (AMP,AMPID(1)),(MINI(1,1),MINY(1)),(D1,DATA(1)),
     1(WORD1,K1(1),ISEC1),(K1(3),IFROM1),(K1(5),IGO1),(WORD2,K2(1),
     2       JSEC,ISEC2),(K2(3),IFROM2),(K2(5),IGO2),(K2(7),NSTAT),
     3(K2(8),L),(WORD3,K3(1),ISEC3),(K3(3),IFROM3),(K3(5),IGO3),
     4(WORDD,KD(1),ISECD),(KD(3),IFROMD),(KD(5),IGOD),
     5(KD(7),NSTAD),(KD(8),LINED)
      DATA AMPID(1),AMPID(2),AMPID(3),AMPID(4),AMPID(5),AMPID(6)
     1/'&','6','5','8','9','1'/
      DATA BLANKS(1),BIN,PEZ/' ','0','.'/
      DATA MON/0,31,59,90,120,151,181,212,243,273,304,334,365,500/
      REWIND 9
      REWIND 10
      ID11=1
      DO 10  I=2,65
   10 BLANKS(I)=BLANKS(1)
      DO 20  I=7,65
   20 AMPIC(I)=BLANKS(1)
      IFROM3 = 1201
      IGO1 = 1202
      ISEC3 = 2000000000
      ISEC1 = - 20000C0C0C
      WORC(1201) = WORD1
      WORC(1202) = WORD3
      DO 30  I=1,16
   30 REAC(5,1050) KSTA(I),STATIO(I)
      JONE(1)=5
      JONE(2)=13
      JONE(3)=18
      JONE(4)=26
      JONE(5)=31
      JONE(6)=39
      JONE(7)=45
```

```
      JCNE(8)=53
      JONE(9)=57
      JONE(10)=65
      NDYR=365
      IPAGE=1
   40 CONTINUE
      READ (5,1090,END=950) INTAPE,INFILE,NDTAPE,NDFILE
      MTOT=C
      M=1
C     CHECK FOR AMPERSAND
  100 READ(9,1000,ERR=100,END=500) DATA
      IF(D1.NE.AMP) GO TO 1C0
C     WHENAMPERSAND HAS BEEN FOUND UPDATE TOTAL
C     NUMBER OF MESSAGES
  110 MTOT=MTOT+1
      NENDAM=0
      L=1
C     READ MESSAGE
  200 READ(9,1000,ERR=200,END=350)(MINI(I,L),I=1,65)
      IF(MINI(1,L).EQ.AMP) GO TO 360
  210 J=1
      J2=1
  300 J1=JCNE(J2)
  310 MTEST=MINI(J,L)-BIN
      IF(MTEST.GT.9) GO TC 34C
      IF(MTEST.LT.0) GO TO 34C
      J=J+1
      IF(J.LT.J1) GO TO 310
C     CHECK FOR PERIOD
      IF(MINI(J,L).NE.PEZ) GO TO 340
      IF(J2.EQ.10) GO TO 320
      J=J+1
      J2=J2+1
      GO TO 3C0
C     ONE GOOD LINE OF DATA HAS BEEN PROCESSED
  320 IF(L.GT.1) GO TO 33C
C     COMPUTE STATION NUMBER FROM FIRST LINE
      NSTAT=MINY(56)-BIN+10*(MINY(55) -BIN)
  322 L = 2
      GO TO 200
  330 DO 334 IK = 1,65
      IF (MINI(IK,L).NE.MINI(IK,1)) GO TO 336
  334 CONTINUE
      GO TO 322
```

```
  336 CONTINUE
      IF ( MINI(56,L).NE.MINY(56)) GO TO 200
      IF ( MINI(55,L).NE.MINY(55)) GO TO 200
      IF (L.GE.33) GO TO 380
C     READ NECT LINE
      L=L+1
      GO TO 200
C     BAD LINE. IF CALIBRATION LINE BAD IGNORE THIS RECORD
  340 CONTINUE
      IF(L.LT.2) GO TO 348
      DO 345 IK = 1,65
      IF(MINI(IK,L).NE.BLANKS(1)) GO TO 200
  345 CONTINUE
      GO TO 370
  348 CONTINUE
      WRITE(6,1070) MTOT
      GO TO 100
C     END OF MESSAGES
  350 NENDAM=1
      GO TO 370
C     NEW MESSAGE AMPERSAND HAS BEEN READ
  360 NENDAM=-1
  370 L=L-1
  380 IF(L.GE.5) GO TO 385
      WRITE (6,1080) MTOT
      GO TO 415
  385 CONTINUE
      IDAY=MINY(107)-BIN+10*(MINY(106)-BIN+10*(MINY(105)-BIN))
      IF(M.LE.1) IDAY1=IDAY
      IDAY=IDAY-IDAY1
      IF(IDAY.GT.180) IDAY=IDAY-NDYR
      IF(IDAY.LT.(-180))  IDAY=IDAY+NDYR
      IH=MINY(93)-BIN+10*(MINY(92)-BIN)
      IM=MINY(80)-BIN+10*(MINY(79)-BIN)
      IS=MINY(67)-BIN+10*(MINY(66)-BIN)
      JSEC=IS+60*(IM+60*(IH+24*IDAY))
      JPLUS = JSEC + 30
      JMINUS = JSEC - 30
      L=65*L
      LI = M
      M = M + 1
  390 IF(JSEC.GE.ISEC3) GO TO 410
      IFROM1 = IFROM3
  400 WORD1 = WORD(IFROM1)
```

```
      IF(JSEC.LT.ISEC1) GO TO 400
      WORD3 = WORD(IGO1)
C
C
      GO TO 420
  410 WORD3 = WORD(IGO3)
      IF(JSEC.GE.ISEC3) GO TO 410
      WORD1 = WORD(IFROM3)
C
C
  420 CONTINUE
      IF(JPLUS.GT.ISEC3) GO TO 2000
  430 IF(JMINUS.LT.ISEC1) GO TO 2020
C
C  NO DUPLICATE MESSAGES
C
  440 CONTINUE
      IFROM2 = IFROM3
      IGO2 = IGO1
      IFROM3 =  LI
      IGO1 = LI
      WORD(IFROM2) = WORD1
      WORD(LI) = WORD2
      WORD(IGO2) = WORD3
      LRECD(LI) = MTOT
      WRITE(11'LI)(MINY(I),I=1,L)
       IF(M.GT.1200) GO TO 500
  415 IF(NENDAM)  110,100,500
  500 CONTINUE
       M = M - 1
      WRITE(6,1010) MTOT,M
      WRITE (6,1100) IPAGE,NDTAPE,INTAPE,NDFILE,INFILE
      N = 0
      WORD2 = WORD(1201)
  510 CONTINUE
      IGO1 = IGO2
      IF(IGO1.GT.1200) GO TO 900
      N = N + 1
      WORD2 = WORD(IGO1)
      DO 810 K=1,16
      IF(NSTAT.EQ.KSTA(K)) GO TO 820
  810 CONTINUE
      K = 16
  820 ITD = ISEC2/86400
```

```
      ITS = ISEC2 - ITD*86400
       IF(ITS.GE.0) GO TO 815
       ITD = ITD - 1
       ITS = ITS + 86400
  815  CONTINUE
      IHR = ITS/3600
      IMIN = (ITS - IHR*3600)/60
      IS = ITS - IHR*3600 - IMIN*60
      ITD=ITD+IDAY1
      IF(ITD.GT.NDYR) ITD=ITD-NDYR
      IF(ITD.LE.0) ITD=ITD+NDYR
      IMON = 2 + ITD/32
      IF (ITD.LE.MON(IMON)) IMON = IMON - 1
      IMD = ITD - MON(IMON)
      READ(11'IG01,ERR=830)(MINY(I),I=1,L)
      L=L/65
      WRITE(6,1020)N,LRECD(IG01),NSTAT,STATIO(K),ITD,IMON,IMD,ITS,
     * IHR,IMIN,IS,L
      WRITE(10,1000) AMPIC
      WRITE(10,1000)((MINI(J,I),J=1,65),I=1,L)
      WRITE(10,1000)  BLANKS
C     WRITE(6,1030)  AMPID
C     WRITE(6,1030) ((MINI(J,I),J=1,65), I=1,L)
C     WRITE(6,1030) BLANKS
      GO TO 510
  830 WRITE(6,1040) N
      GO TO 510
  900 CONTINUE
       END FILE 10
      WRITE(6,1060) IPAGE
      WRITE(6,1060) IPAGE
      GO TO 40
  950 CONTINUE
      REWIND 9
      REWIND 10
      STOP
 1000 FORMAT(65A1)
 1010 FORMAT(15X,'TOTAL NUMBER OF MESSAGES=',I5,
     1 6X,'NUMBER OF PROCESSED MESSAGES='I5)
 1020 FORMAT(6X,I4,14X,I4,16X,I2,12X,A6,12X,I3,5X,I2,5X,I2,7X,I5,5X,I2,
     12X,I2,2X,I2,I5)
 1030 FORMAT(6X,65A1)
 1040 FORMAT(6X,'ERROR RECORD NO.'I5)
 1050 FORMAT(I5,5X,A6)
```

```
 1060 FORMAT(I1)
 1070 FORMAT(6X,'BAD CALIBRATION LINE MESSAGE NO.',I4,X,'REJECTED.')
 1080 FORMAT(6X,'TOO FEW LINES MESSAGE NO.',I4,X,'REJECTED.')
 1090 FORMAT(A8,I5,A8,I5)
 1100 FORMAT(I1,5X,'OUTPUT TAPE',7X,'INPUT TAPE',10X,'STATION',25X,'DAY'
     1,5X,'MONTH',2X,'DAY',6X,'SEC',7X,'HR',2X,'MIN',X,'SEC',X,'LINE'/6X
     1,'NO.',A8,7X,'NO.',A8,9X,'ID',12X,'NAME',60X,'NO.'/6X,'FILE',I5,9X
     1,'FILE',I5/6X,'MESSAGE NO.',7X,'MESSAGE NO.')
 1110 FORMAT(6X,'DUPLICATE MESSAGES',I5,X,'AND',I5,'. THE SECOND MESSAGE
     1 HAS BEEN REJECTED.')
 2000 WORCD = WORD3
 2010 IF(NSTAT.EQ.NSTAD) GO TC 2040
      WORCD= WORD(IGOD)
      IF(JPLUS.GT.ISECD) GO TC 2010
      GC TC 430
 2020 WORCD = WORD1
 2030 IF(NSTAT.EQ.NSTAD) GO TO 2040
      WORCD= WORD(IFROMD)
      IF(JMINUS.LT.ISECD) GO TO 2030
      GO TO 440
 2040 CONTINUE
      M = M - 1
      WORD1 = WORD(IFROMD)
      IF(LINEC.GE.L) GO TO 21C0
      WRITE(6,1110) MTOT,LRECC(IGO1)
      WORD3 = WORD(IGOD)
      LI = IGO1
      IGO1 = IGOD
      IFROM3 = IFROMD
      WORD(IFROM3) = WORD1
      WORD(IGO1) = WORD3
      GO TO 390
 2100 CONTINUE
      WRITE(6,1110) LRECD(IGO1),MTOT
      GO TO 415
      END
/*
//STEP2 EXEC LINKGO
//GO.FT09F001 DD UNIT=(9TRACK,,DEFER),DISP=(OLD,KEEP),LABEL=(7,BLP),
// DCB=(RECFM=FB,BLKSIZE=3250,LRECL=65,DEN=2),VOL=SER=34248B
//GO.FT10F001 DD UNIT=(9TRACK,,DEFER),DISP=(NEW,DELETE),LABEL=(7,BLP),
// DCB=(RECFM=FB,LRECL=65,BLKSIZE=3250,DEN=2),VOL=SER=2135H
//GO.FT11F001 DD UNIT=2314,DSN=&MINI,DISP=(NEW,DELETE),
// DCB=RECFM=FT,SPACE=(CYL,(3C,2))
```

```
//GO.CLEAR DD DSN=&LODMOD(GSFC),DISP=(CLD,DELETE)
//GO.SYSABEND  DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
//              SPACE=(CYL,(1))
//GO.DATA5 DD *
    03      FTMYRS
    05      QUITOE
    06      LIMAPU
    08      SNTAGO
    12      NEWFLD
    15      WNKFLD
    16      JOBURG
    19      ALASKA
    21      CRORAL
    23      MAUGAR
    01      BPOINT
    13      COLEGE
    17      MOJAVE
    14      GRDFKS
    18      WOOMER
    00      NONAME
   34248B     7   2135H     7
/*
                                                                0286 CARDS
```

A-2.3 THE MINITRACK PREPROCESSOR

```
//Z7GEMRWB JOB (G70041150A,T,000138,005005),Z7,MSGLEVEL=1
//   EXEC  LOADER,PARM='MAP,CALL,SIZE=440K',REGION.GO=450K
//GO.SYSLIN    DD UNIT=2400-9,LABEL=(1,BLP),DISP=(OLD,KEEP),
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),VOL=SER=2023G
//GO.FT06F001 DD DUMMY
//GO.FT07F001 DD SYSOUT=B,DSN=DECK
//GO.FT09F001 DD UNIT=2400-9,LABEL=(2,BLP),DISP=(OLD,DELETE),
//   DCB=(RECFM=FB,LRECL=65,BLKSIZE=3250,DEN=2),VOL=SER=1664J
//GO.FT11F001 DD DUMMY
//GO.FT12F001 DD UNIT=2400-9,LABEL=(2,BLP),DISP=(NEW,KEEP),
//   DCB=(RECFM=VBS,LRECL=52,BLKSIZE=5204),VOL=SER=30060D
//GO.FT13F001 DD SYSOUT=A,SPACE=(CYL,(1,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT14F001 DD SYSOUT=A,SPACE=(CYL,(1,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT19F001 DD DUMMY
//GO.FT20F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT21F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT22F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT23F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT24F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT25F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT26F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT27F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT28F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT29F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT30F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT31F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT32F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT33F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
```

```
//GO.FT34F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT35F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT36F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT37F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT38F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT39F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT40F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT41F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT42F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT43F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT44F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT45F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT46F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT47F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT48F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT49F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT50F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.FT51F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//GO.SYSABEND DD SYSOUT=A,SPACE=(CYL,(1))
//GO.DATA5 DD *

FTMYR6 03 369 369    064 28 372 25  957 137   902 00 120 26  013 431     690107
FTM1E 03
FTM2E 03
FTM3E 03
FTM4E 03
```

```
FTM1P 03
FTM2P 03
FTM3P 03
FTM4P 03
QUIT06 05 380 379   952 00 889 00  822 766   082 00 726 00  054 020    690107
       05
       05
       05
       05
       05
       05
       05
       05
LIMAP6 06 382 382   460 28 655 52  950 022   940 00 966 00  247 876    690107
       06
       06
       06
       06
       06
       06
       06
       06
SNTAG6 08 391 391   983 00 106 00  078 938   013 00 976 00  042 970    690107
       08
       08
       08
       08
       08
       08
       08
       08
NEWFL6 12 372 372   467 29 070 25  935 174   887 00 518 28  929 914    690107
       12
       12
       12
       12
       12
       12
       12
       12
WNKEL6 15 385 384   540 29 222 25  072 872   100 00 791 28  975 012    690107
       15
       15
       15
```

	15														
	15														
	15														
	15														
	15														
JOBUR6	16	413	412	418-29	186	00	053	094	575-29	835	00	938	870	690107	
	16														
	16														
	16														
	16														
	16														
	16														
	16														
	16														
ULASK6	19	362	363	660 00	967	00	942	450	069 00	490	00	076	927	690107	
	19														
	19														
	19														
	19														
	19														
	19														
	19														
	19														
ORORA6	21	364	364	010 00	785	00	783	008	775 00	303	00	121	933	690107	
	21														
	21														
	21														
	21														
	21														
	21														
	21														
	21														
MADGA6	23	413	412	676 00	030	00	047	015	039 00	856	00	884	154	690107	
	23														
	23														
	23														
	23														
	23														
	23														
	23														
	23														
BPOIN6	01	363	363	743 00	223	00	948	083	255 00	933	00	094	889	690107	
	01														
	01														

```
       01
       01
       01
       01
       01
       01
COLEG6 13 381 381   668 00 624 00  959 842   253 00 011 00  807 926    690107
       13
       13
       13
       13
       13
       13
       13
       13
MOJAV6 17 375 375   009 00 844 00  030 076   067 00 169 00  577 126    690107
       17
       17
       17
       17
       17
       17
       17
       17
GFORK6 14 370 370   481 00 161 00  064 603   976 00 704 00  949 002    690107
       14
       14
       14
       14
       14
       14
       14
       14
WOMER6 18 395 395   945 00 109 00  031 950   173 00 033 00  935 120    690107
       18
       18
       18
       18
       18
       18
       18
       18
CHAOS1 99
       99
```

```
     99
     99
     99
     99
     99
     99
     99
65891              136.830
00000
/*
//GO.FT08F001 DD *
   GEOS-A MINITRACK DATA STARTING 08 JAN 1966    ARC 3
    1  217   66    2    1

/*
//GO.FT08F002 DD DUMMY
//GO.FT08F003 DD DUMMY
//GO.FT08F004 DD DUMMY
/*
                                                                0239 CARDS
```

```
      INTEGER SIG1,SIG2,SIG3,SIG4,SIG5,INO,ISO
      INTECER AMP,END,CSTA,CANT,SIG,ANTD,HORD,MIND,SECD,SAT,DATE,KFA,KFB
      INTEGER IGRADE
      REAL NSFD,NSCD,NSMD
      REAL IDIF1,IDIF2,IDIF3,IDIF4,IDIF5
      REAL NSM,NSC,NSFPO,NSFEC,IDIF,     NSF1,NSF2,NSF3,NSF4,NSF5
      LOGICAL*1 SLE,SLN
      LOGICAL*1 ASTA,ARMODA,ACUR,IALOBE,IARATE,IAACC,IAWMER,IAWCER,IBLOB
     XE,IBRATE,IBACC,IBNMER,IBNCER,ICIGA,IOUR,ISEC,LCOS,MCOS,IENO,INNO
      LOGICAL*1 DATA,PEZ,BIN,SPX,IAMP
      LOGICAL*1 MSGR
      LOGICAL *1 LFILE
      REAL*8 XNAP
      REAL*8 STATIO
      REAL*8 DNAP
      REAL*8 A,A1,A2,A3
      DIMENSION STATIO(34),KFA(17),KFB(17),EWM(17),CLEWM(17),EWC(17),
     1  CLEWC(17), EWFEQ(17), NSM(17), CLNSM(17), NSC(17), CLNSC(17),
     2  NSFEQ(17), NSFPO(17), ISTA(68), IANT(68), C1(68), C2(68),C3(68),
     3  C4(68), C5(68), C6(68), C7(68), C8(68), KSAID(50), FREQ(50),
     4  KSTA(48), EWFPO(17), TIM(31), CO(68)
      DIMENSION SECD(31),EWMD(31),EWCD(31),EWFD(31),NSMD(31),NSCD(31),NS
     1FD(31),MIND(31),HORD(31),DAYD(31),ANTD(31),STAD(31),SIGD(31),EEWF
     2(31),ENSF(31),IDAYD(31),AST(17),DATE(17),CD(68)
      DIMENSION ASTA(3),ARMODA(7),AOUR(7),IALOBE(6),IARATE(5),IAACC(4),I
     XAWMER(3),IAWCER(3),IBLOBE(6),IBRATE(5),IBACC(4),IBNMER(3),IBNCER(3
     X),ICIGA(3),IOUR(5),ISEC(6),LCOS(8),MCOS(8),IENO(3),INNO(3)
      DIMENSION MSGR(80)
      DIMENSION DATA(100)
      DIMENSION SLE(4),SLN(4)
      DIMENSION TFOVEW(31,5),FOVEW(31,5),TFOVNS(31,5),FOVNS(31,5)
      DIMENSION DNAP(6),JUDY(34),IPASS(34),JUSEC(34)
      DIMENSION XNAP(6)
      DIMENSION IA(17)
       DIMENSION IPASST(34)
      EQUIVALENCE (SLN(1),INOVER)
      EQUIVALENCE (SLE(1),IEOVER)
      DIMENSION AACOS(31,5),BBCOS(31,5)
      DATA PER,ASK,SPA,POL,EQ,F1,F2,F3,F4,F5,F6,F7,IAMP,SLA,PEZ,BIN,SPX/
     1Z4B404040,Z5C404040,Z40404040,ZD7404040,ZC5404040,ZC1404040,ZC2404
     2040,ZC3404040,ZC4404040,ZC5404040,ZC6404040,ZC7404040,Z50,Z6140404
```

```
      30,Z4B,ZF0,Z40/
       DATA STATIO(17)/'NONAME'/
       DATA A1,A2,A3/'PASS','JUDAY','JUSEC'/
       SLA=SLA
       KDEG=25
       LFILE=.FALSE.
       DEGREE=KDEG
       CCOS=COS(DEGREE*3.1459265/180.)
       KSTA(17)=0
       XNAP(5)=0.D0
       XNAP(6)=0.D0
       DNAP(6)=0.D0
       DNAP(5)=0.D0
       DO 330 I=1,34
       IPASS(I)=0
        IPASST(I) = 0
       JUDY(I)=0
   330 JUSEC(I)=0
       READ( 5,760)IGRADE
   760 FORMAT(9X,I1)
       JL=0
       JM=0
       WRITE( 6,760)IGRADE
  1340 FORMAT (1H1,19('*'),80X,20('*')
      *         //50X,'INPUT STATION CONSTANTS'//
      *         2X,'STATION    KFA  KFB  EWM CLEWM  EWC  CLEWC  EWFEQ',
      *         1X,'EWFPO  NSM  CLNSM  NSC  CLNSC  NSFEQ  NSFPO  DATE'/
      *         2X,'NAME  NO.'//)
       WRITE (13,1340)
  1341 FORMAT (1X,A6,1X,I3,2(2X,I3),1X,F4.3,2X,F4.3,2X,F4.3,1X,F4.3,
      *         2(3X,F4.3,2X,F4.3),1X,F4.3,2X,F4.3,2(3X,F4.3),3X,I6)
       DO 35 J=1,16
C      INPUT STATION CONSTANTS
       READ( 5,80) STATIO(J),KSTA(J),KFA(J),KFB(J),EWM(J),CLEWM(J),EWC(J)
      1,CLEWC(J),EWFEQ(J),EWFPC(J),NSM(J),CLNSM(J),NSC(J),CLNSC(J),NSFEQ(
      2J),NSFPO(J),DATE(J)
       WRITE(6,580)STATIO(J),KSTA(J),KFA(J),KFB(J),EWM(J),CLEWM(J),EWC(J)
      1,CLEWC(J),EWFEQ(J),EWFPC(J),NSM(J),CLNSM(J),NSC(J),CLNSC(J),NSFEQ(
      2J),NSFPO(J),DATE(J)
    80 FORMAT(A6,X,I2,I4,I4,3X,F4.3,F3.3,F4.3,F3.3,X,F4.3,F4.3,3X,F4.3,F3
      X.3,F4.3,F3.3,X,F4.3,F4.3,5X,I6)
   580 FORMAT(X,A6,X,I2,X,I3,X,I3,3X,F4.3,F4.3,F4.3,F4.3,X,F4.3,F4.3,
      X3X,F4.3,F4.3,F4.3,F4.3,X,F4.3,F4.3,2X,I6)
       WRITE (13,1341) STATIO(J),KSTA(J),KFA(J),KFB(J),EWM(J),CLEWM(J),
```

```
     *       EWC(J),CLEWC(J),EWFEQ(J),EWFPU(J),NSM(J),CLNSM(J),
     *       NSC(J),CLNSC(J),NSFEQ(J),NSFPO(J),DATE(J)
      DO 36 M=1,4
      JL=JM+M
C     INPUT STATION COEFF.
      READ( 5,81) IANT(JL),ISTA(JL),CO(JL),C1(JL),C2(JL),C3(JL),C4(JL)
      WRITE(6,811)IANT(JL),ISTA(JL),CO(JL),C1(JL),C2(JL),C3(JL),C4(JL)
  811 FORMAT(4X,A1,X,I2,5(X,E15.8))
      READ(5,581) C5(JL),C6(JL),C7(JL),C8(JL)
  581 FORMAT(8X,4(X,E12.8))
      WRITE(6,681)C5(JL),C6(JL),C7(JL),C8(JL)
  681 FORMAT(9X,4(X,E15.8))
   81 FORMAT(4X,A1,X,I2,5(X,E12.8))
   36 CONTINUE
      JM=JM+4
   35 CONTINUE
      WRITE(6,500)(KSTA(II),II=1,16)
  500 FORMAT(X, 16(X,I2))
      DO 2 I=1,17
      STATIO(I+17)=STATIO(I)
      KSTA(I+17)=KSTA(I)+100
    2 CONTINUE
      DO 37 J=1,50
C     INPUT SATELLITE CONSTANTS
      READ( 5,82) KSAID(J),FREQ(J)
   82 FORMAT(I5,19X,F8.3)
      WRITE( 6,582) KSAID(J),FREQ(J)
  582 FORMAT(X,I5,19X,F8.3)
      IF(KSAID(J))37,38,37
   37 CONTINUE
   38 KSATCT=J-1
      READ(8,305,END=321,ERR=902)(MSGR(I),I=1,80)
  305 FORMAT(80A1)
       NAPEND = 0
       IB1 = 0
       REWIND 9
       IARC = 1
       IARCS=-1
       I201 = 201
       I202 = 202
      WRITE(13,340)(MSGR(I),I=1,80)
  340 FORMAT(1H1,19('*'),80A1,20('*')//50X,'PROCESSING MESSAGES'/)
      WRITE(14,340)(MSGR(I),I=1,80)
      WRITE(14,352)
```

```
352 FORMAT (3X,'SUMMARY OF DATA MESSAGES PRODUCED'//3X,'MESSAGE',1X,    000130
   *       'DATA',3X,'STATION',3X,'ARC',2X,'PASS',2X,'START TIME',      000131
   *       6X,'L',9X,'M',9X,'N',7X,'LDOT',6X,'MDOT',6X,'NDOT'           000132
   *       / 3X,'NUMBER',3X,'PTS',2X,'NO.',2X,'NAME',14X,'DAY',2X,      000133
   *       'SEC.'//)                                                     000134
902 CONTINUE                                                             000135
     IF(NAPEND) 910,920,3                                                000136
910  READ(9,601,END=915,ERR=910) (DATA(I),I=1,65)                        000137
     GO TO 910                                                           000138
915  IB1 = 0                                                             000139
920  CONTINUE                                                            000140
     READ(8,901,END=1,ERR=920) ILOW,IHIGH,IYEAR,NEWARC,NAPEND            000141
     IF(NEWARC.LE.0) GO TO 2010                                          000142
     IARC = NEWARC                                                       000143
     IF(IARCS.EQ.IARC) GO TO 2010                                        000144
     DO 2005 I= 1,24                                                     000145
      IPASST(I)=IPASS(I)+IPASST(I)                                       000146
2005 IPASS(I) = 0                                                        000147
2010  CONTINUE                                                           000148
    IF(ILOW.GT.IHIGH)IHIGH=ILOW                                          0001490
312 FORMAT(A5,13I5)                                                      0001500
    K=0                                                                  0001510
324 READ(8,312,ERR=313,END=313)A,N,(IA(I),I=1,12)                        0001520
     READ(8,901,ERR=313,END=313)  (IA(I), I=13,17)                       000153
    K=K+1                                                                0001540
    IF(A.EQ.A1)GO TO 314                                                 0001550
    IF(A.EQ.A2)GO TO 315                                                 0001560
    IF(A.EQ.A3)GO TO 316                                                 0001570
    GO TO 313                                                            0001580
314 DO 317 I=1,17                                                        000159
317 IPASS(I+N-1)=IA(I)                                                   0001600
    GO TO 318                                                            0001610
315 DO 319 I=1,17                                                        000162
319 JUDY(I+N-1)=IA(I)                                                    0001630
    GO TO 318                                                            0001640
316 DO 323 I=1,17                                                        000165
323 JUSEC(I+N-1)=IA(I)                                                   0001660
318 IF(K.LT.6)GO TO 324                                                  0001670
313 CONTINUE                                                             0001680
    IPAGE=1                                                              0001690
    WRITE(6,302)IPAGE,(MSGR(I),I=1,80)                                   0001700
302 FORMAT(I1,19X,80A1,/)                                                0001710
    IPAGE=0                                                              0001720
    WRITE(6,903)ILOW,IHIGH                                               0001730
```

```
  903 FORMAT('ODATA SET RANGES FROM DATA FILE    LOW = ',I5,4X,'HIGH = '
     1,I5)
      KEND=0
  901   FORMAT(5I5)
       IF(IB1.LT.ILOW) GO TO 39
       IF(IB1.EQ.ILOW) GO TO 2001
       IB1=0
       REWIND 9
C     READ SATELLITE IDENTIFICATION,CHECK FOR AMPERSAND****************
   39 CONTINUE
      READ(9,601,END=902,ERR=39)(DATA(I),I=1,65)
  601 FORMAT(65A1)
   94 CONTINUE
      IF(DATA(1).NE.IAMP)GO TO 39
      IF(DATA(7).EQ.SPX)GO TO 83
      IF(DATA(8).EQ.SPX)GO TO 83
      IF(DATA(4)-BIN.EQ.0)GO TO 85
      SAT=((DATA(3)-BIN)*10000.)+((DATA(4)-BIN)*1000.)+((DATA(5)-BIN)*10
     X0.)+((DATA(6)-BIN)*10.)+(DATA(8)-BIN)
      GO TO 84
   85 SAT=((DATA(2)-BIN)*10000.)+((DATA(3)-BIN)*1000.)+((DATA(5)-BIN)*10
     X0.)+((DATA(6)-BIN)*10.)+(DATA(8)-BIN)
      IGRADE=(DATA(10)-BIN)
      IYEAR=(DATA(12)-BIN)*10+(DATA(13)-BIN)
      GO TO 84
   83 SAT=((DATA(2)-BIN)*10000.)+((DATA(3)-BIN)*1000.)+((DATA(4)-BIN)*10
     X0)+((DATA(5)-BIN)*10)+(DATA(6)-BIN)
   84 CONTINUE
      IB1=IB1+1
      IF(IB1.LT.ILOW)GO TO 39
      IF(IB1.GT.IHIGH)GO TO 902
 2001  CONTINUE
      DO 602 M=1,50
      IF(SAT.EQ.KSAID(M))GO TO 603
  602 CONTINUE
      WRITE( 6,635)
  635 FORMAT(19H SAID NOT IN TABLE )
      WRITE(6,183)IAMP,SAT
  183 FORMAT(X,A1,I5)
      GO TO 604
  603 CONTINUE
      IF(IPAGE.EQ.0)GO TO 320
      WRITE(6,302)IPAGE,(MSGR(I),I=1,80)
      IPAGE=0
```

```
  320 WRITE(6,310)IPAGE,IB1
      WRITE(6,660)(DATA(I),I=1,65)
      KKSAID=KSAID(M)
C     READ CAL.LINE WITH FORMAT CHECK *******************************
      READ(9,601,END=902,ERR=30)(DATA(I),I=1,65)
      IF(DATA(1).EQ.IAMP)GO TO 83
      WRITE(6,660)(DATA(I),I=1,65)
C     CHECK PERIODS IN CAL. LINE
      IF(DATA(5).NE.PEZ)GO TO 604
      IF(DATA(13).NE.PEZ)GO TO 604
      IF(DATA(18).NE.PEZ)GO TO 604
      IF(DATA(26).NE.PEZ)GO TO 604
      IF(DATA(31).NE.PEZ)GO TO 604
      IF(DATA(39).NE.PEZ)GO TO 604
      IF(DATA(45).NE.PEZ)GO TO 604
      IF(DATA(53).NE.PEZ)GO TO 604
      IF(DATA(57).NE.PEZ)GO TO 604
      IF(DATA(65).NE.PEZ)GO TO 604
      DO 605 K=1,4
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  605 CONTINUE
      DO 606 K=6,12
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  606 CONTINUE
      DO 607 K=14,17
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  607 CONTINUE
      DO 608 K=19,25
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  608 CONTINUE
      DO 609 K=27,30
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  609 CONTINUE
      DO 610 K=32,38
```

```
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  610 CONTINUE
      DO 611 K=40,44
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  611 CONTINUE
      DO 612 K=46,52
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  612 CONTINUE
      DO 613 K=54,56
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  613 CONTINUE
      DO 614 K=58,64
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  614 CONTINUE
      X=DATA(9)+DATA(22)+DATA(35)+DATA(49)+DATA(61)
      IF(X.NE.45.)GO TO 604
      CSTA=(DATA(55)*10.)+(DATA(56))
      DO 616 L=1,17
      IF(KSTA(L).EQ.CSTA)GO TO 617
  616 CONTINUE
      WRITE( 6,618)
  618 FORMAT(27H WRONG STATION IN CAL.LINE )
      GO TO 604
  617 D=DATA(3)*10+DATA(4)
      CEWM=D/100.
      D=DATA(16)*10+DATA(17)
      CEWC=D/100.
      D=DATA(6)*100+DATA(7)*1C+DATA(8)
      CEWF1=D/1000.
      D=DATA(19)*100+DATA(20)*10+DATA(21)
      CEWF2=D/1000.
      D=DATA(32)*100+DATA(33)*10+DATA(34)
      CEWF3=D/1000.
      D=DATA(46)*100+DATA(47)*10+DATA(48)
```

```
      CEWF4=D/1000.
      D=DATA(58)*100+DATA(59)*10+DATA(60)
      CEWF5=D/1000.
      D=DATA(10)*100+DATA(11)*10+DATA(12)
      CNSF1=D/1000.
      D=DATA(23)*100+DATA(24)*10+DATA(25)
      CNSF2=D/1000.
      D=DATA(36)*100+DATA(37)*10+DATA(38)
      CNSF3=D/1000.
      D=DATA(50)*100+DATA(51)*10+DATA(52)
      CNSF4=D/1000.
      D=DATA(62)*100+DATA(63)*10+DATA(64)
      CNSF5=D/1000.
      D=DATA(29)*10+DATA(30)
      CNSM=D/100.
      D=DATA(43)*10+DATA(44)
      CNSC=D/100.
      CSTA=DATA(55)*10+DATA(56)
       I11 = 1
       WRITE (11,310) I11,IB1
      WRITE(11,501)IAMP,SAT,CSTA,STATIO(L)
  501 FORMAT(X,A1,I5,X,I2,X,A6)
      CANT=DATA(54)
      END=DATA(65)
   43 CEWM=EWM(L)+CEWM
      CEWC=EWC(L)+CEWC
      CNSM=NSM(L)+CNSM
      CNSC=NSC(L)+CNSC
      KM=5.
      RATE=0.
      IDIF1=CEWF2-CEWF1
      IDIF2=CEWF3-CEWF2
      IDIF3=CEWF4-CEWF3
      IDIF4=CEWF5-CEWF4
      CEWF1=CEWF3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35.)
      IDIF=0.
      IF(CANT-2.)162,160,161
C     NARROW BAND TRACKING FILTER ****************************************
  160 IDIF=.0
      AST(1)=SPA
      GO TO 164
  161 IDIF=.0
      AST(1)=ASK
      GO TO 164
```

```
  162 IF(CANT.EQ.1.)GO TO 164
  163 IDIF=.120
C     2 CPS TRACKING FILTER ******************************************
      AST(1)=PER
  164 IDIF1=CNSF2-CNSF1
      IDIF2=CNSF3-CNSF2
      IDIF3=CNSF4-CNSF3
      IDIF4=CNSF5-CNSF4
      CNSF1=CNSF3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35.)
C     CABLE LENGTH INEQUALITIES
      CEWM=((CLEWM(L)/.846)*(136.5-FREQ(M)))+CEWM
      CEWC=((CLEWC(L)/.846)*(136.5-FREQ(M)))+CEWC
      CNSM=((CLNSM(L)/.846)*(136.5-FREQ(M)))+CNSM
      CNSC=((CLNSC(L)/.846)*(136.5-FREQ(M)))+CNSC
      WRITE(11,636)
  636 FORMAT(55H CALIBRATED PHASE READINGS,5 POINT FITTED FINE READINGS)
C     CALIBRATED ZENITH **********************************************
      WRITE(11,639)
  639 FORMAT(63H   CEWM        CEWC        CNSM        CNSC        CEWF
     X  CNSF )
      WRITE(11,150)CEWM,CEWC,CNSM,CNSC,CEWF1,CNSF1
  150 FORMAT(6(X,F10.6))
      WRITE(11,640)
  640 FORMAT(74H HRMNSC      EWFINE      EWMEDM      EWCORS      NSFINE
     XNSMEDM     NSCORS )
C     READ DATA WITH FORMAT CHECK************************************
      DO 41 JK=1,60
      K=JK
      IF(K.EQ.32.)GO TO 120
  621 READ(9,601,END=95,ERR=32)(DATA(I),I=1,65)
      IF(DATA(1).EQ.IAMP)GO TO 120
      WRITE(6,660)(DATA(I),I=1,65)
      IF(DATA(10).EQ.SPX)GO TO 122
      GO TO 124
  122 IF(DATA(30).EQ.SPX)GO TO 123
      GO TO 124
  123 IF(DATA(50).EQ.SPX)GO TO 120
  124 IF(DATA(5).NE.PEZ)GO TO 621
C     CHECK PERIODS IN DATA LINE
      IF(DATA(13).NE.PEZ)GO TO 621
      IF(DATA(18).NE.PEZ)GO TO 621
      IF(DATA(26).NE.PEZ)GO TO 621
      IF(DATA(31).NE.PEZ)GO TO 621
      IF(DATA(39).NE.PEZ)GO TO 621
```

```
      IF(DATA(45).NE.PEZ)GO TC 621
      IF(DATA(53).NE.PEZ)GO TC 621
      IF(DATA(57).NE.PEZ)GO TO 621
      IF(DATA(65).NE.PEZ)GO TO 621
      DO 622 J=1,4
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  622 CONTINUE
      DO 623 J=6,12
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  623 CONTINUE
      DO 624 J=14,17
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  624 CONTINUE
      DO 625 J=19,25
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  625 CONTINUE
      DO 626 J=27,30
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  626 CONTINUE
      DO 627 J=32,38
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  627 CONTINUE
      DO 628 J=40,44
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  628 CONTINUE
      DO 629 J=46,52
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  629 CONTINUE
```

```
      DO 630 J=54,56
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  630 CONTINUE
      DO 631 J=58,64
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  631 CONTINUE
      SECD(K)=DATA(1)*10+DATA(2)
      MIND(K)=DATA(14)*10+DATA(15)
      HORD(K)=DATA(27)*10+DATA(28)
      ANTD(K)=DATA(54)
      IDAYD(K)=DATA(40)*100+DATA(41)*10+DATA(42)
      STAD(K)=DATA(55)*10+DATA(56)
      D=DATA(3)*10+DATA(4)
      EWMD(K)=D/100.
      D=DATA(16)*10+DATA(17)
      EWCD(K)=D/100.
      D=DATA(29)*10+DATA(30)
      NSMD(K)=D/100.
      D=DATA(43)*10+DATA(44)
      NSCD(K)=D/100.
      D=DATA(6)*100+DATA(7)*1C+DATA(8)
      EWF1=D/1000.
      D=DATA(19)*100+DATA(20)*10+DATA(21)
      EWF2=D/1000.
      D=DATA(32)*100+DATA(33)*10+DATA(34)
      EWF3=D/1000.
      D=DATA(46)*100+DATA(47)*10+DATA(48)
      EWF4=D/1000.
      D=DATA(58)*100+DATA(59)*10+DATA(60)
      EWF5=D/1000.
      D=DATA(10)*100+DATA(11)*10+DATA(12)
      NSF1=D/1000.
      D=DATA(23)*100+DATA(24)*10+DATA(25)
      NSF2=D/1000.
      D=DATA(36)*100+DATA(37)*10+DATA(38)
      NSF3=D/1000.
      D=DATA(50)*100+DATA(51)*10+DATA(52)
      NSF4=D/1000.
      D=DATA(62)*100+DATA(63)*10+DATA(64)
      NSF5=D/1000.
```

```
      SIG1=DATA(9)
      SIG2=DATA(22)
      SIG3=DATA(35)
      SIG4=DATA(49)
      SIG5=DATA(61)
      END=DATA(65)
  100 IF(ANTD(K)-2)102,101,101
  101 ANT=57.
      GO TO 103
  102 ANT=46.
  103 TIM(K)=((HORD(K)*3600.)+(60.*MIND(K)))+SECD(K)
      FOVEW(K,1)=EWF1
      FOVEW(K,2)=EWF2
      FOVEW(K,3)=EWF3
      FOVEW(K,4)=EWF4
      FOVEW(K,5)=EWF5
      FOVNS(K,1)=NSF1
      FOVNS(K,2)=NSF2
      FOVNS(K,3)=NSF3
      FOVNS(K,4)=NSF4
      FOVNS(K,5)=NSF5
C     FIT FIVE FINES EACH LINE
      IDIF1=EWF2-EWF1
      CALL NORMAL(IDIF1)
      IDIF2=EWF3-EWF2
      CALL NORMAL(IDIF2)
      IDIF3=EWF4-EWF3
      CALL NORMAL(IDIF3)
      IDIF4=EWF5-EWF4
      CALL NORMAL(IDIF4)
      EEWF(K)=EWF3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35.)
      IDIF5=((IDIF1+IDIF2+IDIF3+IDIF4)/4.)
       ICTEW = IDIF5 * 0.05
C     COUNTER DELAY (TIME) ******************************************
      EEWF(K)=(EEWF(K)-(.05*IDIF5*EWF3))
      IDIF1=NSF2-NSF1
      CALL NORMAL(IDIF1)
      IDIF2=NSF3-NSF2
      CALL NORMAL(IDIF2)
      IDIF3=NSF4-NSF3
      CALL NORMAL(IDIF3)
      IDIF4=NSF5-NSF4
      CALL NORMAL(IDIF4)
      ENSF(K)=NSF3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35.)
```

```
      IDIF5=((IDIF1+IDIF2+IDIF3+IDIF4)/4.)
       ICTNS = IDIF5 * 0.05
       TIM2 = TIM(K) - TIM(1) + 0.000001
       DO 300 IFOV = 1,5
      TFOVEW(K,IFOV)=TIM2+0.2*(IFOV-1)+FOVEW(K,IFOV)*0.01
      TFOVNS(K,IFOV)=TIM2+0.2*(IFOV-1)+FOVNS(K,IFOV)*0.01
  300  CONTINUE
C     FILTER DELAY (TIME)
      ENSF(K)=ENSF(K)-(.05*IDIF5*NSF3)
      IDIF1=SIG2-SIG1
      IDIF2=SIG3-SIG2
      IDIF3=SIG4-SIG3
      IDIF4=SIG5-SIG4
      SIGD(K)=SIG3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35)
      WRITE(11,151)HORD(K),MIND(K),SECD(K),EEWF(K),EWMD(K),EWCD(K),ENSF(
     XK),NSMD(K),NSCD(K)
  151 FORMAT(X,I2,I2,I2,X,(6(F10.6,X)))
   41 CONTINUE
      GO TO 120
   95 CONTINUE
      KEND=1
C     DATA MSG.COMPLETE,START SMOOTHING *******************************
  120 SL=.01
      AST(6)=SPA
      K=K-1
      NBRK=K
      IF(K.LE.5)GO TO 780
      GO TO 782
  780 WRITE( 6,781)
  781 FORMAT(27H LESS THAN 5 LINES OF DATA )
      KZ=K-1
      WRITE(13,341)IB1,CSTA,STATIO(L)
      WRITE(13,346)
  346 FORMAT(1H+,32X,'MESSAGE TOO SHORT')
      IF(KEND.EQ.1)GO TO 902
      GO TO 78
  782 JK=K
      IF(ANT.EQ.57.) GO TO 90
      GO TO 91
   90 CEWF1=CEWF1+EWFPO(L)
      CNSF1=CNSF1+NSFPO(L)
      AST(2)=POL
      GO TO 121
   91 CEWF1=CEWF1+EWFEQ(L)
```

```
      CNSF1=CNSF1+NSFEQ(L)
      AST(2)=EQ
  121 SM=2.0
      WRITE(11,638)
  638 FORMAT(32H KC-KS1 USING STATION CONSTANTS )
      WRITE(11,1152)CEWF1,EWFPO(L),EWFEQ(L),CNSF1,NSFPO(L),NSFEQ(L)
C     CHECK TIME SEQUENCE ***************************************
      IB=0
      IC=0
      ID=0
      IE=0
      IF=0
      IG=0
      IH=0
      KQ=K-1
      DO 20 KS=1,KQ
  806 ITZM=TIM(KS+1)-TIM(KS)
      IF(ITZM)807,807,808
  807 TIM(KS+1)=TIM(KS+1)+86400.
      GO TO 806
  808 IF(ITZM.NE.1)GO TO 21
      IB=IB+1
      GO TO 20
   21 IF(ITZM.NE.2 )GO TO 22
      IC=IC+1
      GO TO 20
   22 IF(ITZM.NE.10 )GO TO 23
      ID=ID+1
      GO TO 20
   23 IF(ITZM.NE.20 )GO TO 24
      IE=IE+1
      GO TO 20
   24 IF(ITZM.NE.60 )GO TO 25
      IF=IF+1
      GO TO 20
   25 IF(ITZM.NE.120 )GO TO 26
      IG=IG+1
      GO TO 20
   26 IF(ITZM.NE.600 )GO TO 20
      IH=IH+1
      GO TO 20
   20 CONTINUE
      ITZM=MAXO(IB,IC,ID,IE,IF,IG,IH)
      IF(IB.EQ.ITZM)GO TO 200
```

```
      IF(IC.EQ.ITZM)GO TO 201
      IF(ID.EQ.ITZM)GO TO 202
      IF(IE.EQ.ITZM)GO TO 203
      IF(IF.EQ.ITZM)GO TO 204
      IF(IG.EQ.ITZM)GO TO 205
      IF(IH.EQ.ITZM)GO TO 206
  200 AST(3)=F1
      TILI=1.
      GO TO 207
  201 AST(3)=F2
      TILI=2.
      GO TO 207
  202 AST(3)=F3
      TILI=10.
      GO TO 207
  203 AST(3)=F4
      TILI=20.
      GO TO 207
  204 AST(3)=F5
      TILI=60.
      GO TO 207
  205 AST(3)=F6
      TILI=120.
      GO TO 207
  206 AST(3)=F7
      TILI=600.
      GO TO 207
   72 WRITE( 6,73)
   73 FORMAT(22H TIME OUT OF SEQUANCE )
      WRITE(13,341)IB1,CSTA,STATIO(L)
      WRITE(13,345)
  345 FORMAT(1H+,23X,'FRAME TIME OUT OF SEQUENCE')
      KZ=K-1
   78 WRITE( 6,77)SAT,CSTA,STATIO(L),HORD(KZ),MIND(KZ),IDAYD(KZ)
   77 FORMAT(X,I5,X,I2,X,A6,X,I2,I2,X,I3)
      GO TO 65
   74 WRITE( 6,75)
   75 FORMAT(25H DATA EXCEEDS TIME CHECK )
      WRITE(13,341)IB1,CSTA,STATIO(L)
      WRITE(13,347)
  347 FORMAT(1H+,31X,'EXCEEDS TIME CHECK')
      KZ=K-1
      GO TO 78
   50 WRITE(11,51)
```

```
   51 FORMAT(40H EAST MEDIUM CHANNEL EXCEEDS 100 COUNTS )
      AST(6)=F1
      RATE=0.
      GO TO 58
   52 WRITE(11,53)
   53 FORMAT(40H EAST COURSE CHANNEL EXCEEDS 100 COUNTS )
      AST(6)=F1
      RATE=0.
      GO TO 58
   54 WRITE(11,55)
   55 FORMAT(41H NORTH MEDIUM CHANNEL EXCEEDS 100 COUNTS )
      AST(6)=F2
      RATE=0.
      GO TO 59
   56 WRITE(11,57)
   57 FORMAT(41H NORTH COURSE CHANNEL EXCEEDS 100 COUNTS )
      AST(6)=F2
      RATE=0.
      GO TO 59
  804 WRITE(11,805)
  805 FORMAT(27H DATA WILL NOT LOBE ASSIGN )
      WRITE(13,341)IB1,CSTA,STATIO(L)
      WRITE(13,348)
  348 FORMAT(1H+,29X,'WILL NOT LOBE ASSIGN')
      KZ=K-1
      GO TO 78
C     EW AMBIGUITY LOBE ASSIGN.
  207 SL=.015
      K=JK-1
      DO 70 N=1,K
      IF(TIM(N+1)-TIM(N))72,72,71
   71 IF(TIM(N+1)-TIM(N)-(5.*TILI))70,70,74
   70 CONTINUE
      K=JK
      CALL LOBASN(TIM,K,EWMD,RATE,ITD)
      IF(ITD.GE.100)GO TO 804
      CALL LOBASN(TIM,K,EWCD,RATE,ITD)
      IF(ITD.GE.100)GO TO 804
      CALL LSQQUA(TIM,ALPHA,EWMD,K,MID,SIA,SL,EWMB,EWMC,SM,ENA,EOA)
      EWMA=ALPHA+EWMD(MID)
      EWMT=TIM(MID)-.15
      SA=SIA
      WRITE(11,643)
  643 FORMAT(83H         ALPHA.EWM    MID.PT.    RATE          SIGMA
```

```
     X    BETA              GAMMA    )
      WRITE(11,152)ALPHA,MID,RATE,SIA,EWMB,EWMC
  152 FORMAT(X,F14.6,X,I6,X,4(F14.6,X))
      K=JK
      CALL LSQQUA(TIM,ALPHA,EWCD,K,MID,SIA,SL,EWCB,EWCC,SM,ENB,EOB)
      EWCA=ALPHA+EWCD(MID)
      EWCT=TIM(MID)+.05
      SB=SIA
      WRITE(11,644)
  644 FORMAT(83H            ALPHA.EWC   MID.PT.   RATE              SIGMA
     X    BETA              GAMMA    )
      WRITE(11,152)ALPHA,MID,RATE,SIA,EWCB,EWCC
      RATE=((EWMB*ANT/4.)+(EWCB*ANT/3.5))/2.
      IF(ABS(RATE).LE..05)RATE=0.0
C     EWF LOBE ASSIGN.
      IF(ABS(SB).GT..1)GO TO 52
      IF(ABS(SA).GT..1)GO TO 50
   58 SL=.01
      SM=2.5
      K=JK
      CALL LOBASN(TIM,K,EEWF,RATE,ITD)
      IF(ITD.GE.100)GO TO 804
      CALL LSQQUA(TIM,ALPHA,EEWF,K,MID,SIA,SL,EWFB,EWFC,SM,ENC,EOC)
      MIDEW=MID
      EWFT=TIM(MID)
      EWFA=ALPHA+EEWF(MID)
  224 SC=SIA
      INO=SIA*1000.+.5
  227 CALL ZERO(3,INO,IENO)
      IDAYD(1)=IDAYD(MID)
      WRITE(11,642)
  642 FORMAT(83H            ALPHA.EWF   MID.PT.   RATE              SIGMA
     X    BETA              GAMMA    )
      WRITE(11,152)ALPHA,MID,RATE,SIA,EWFB,EWFC
      WRITE(11,153)EWMA,EWMT,EWCA,EWCT,RATE
  153 FORMAT(6(F14.6,X))
C     NS AMBIGUITY LOBE ASSIGN.
      SL=.015
      RATE=0.
      K=JK
      SM=2.0
      CALL LOBASN(TIM,K,NSMD,RATE,ITD)
      IF(ITD.GE.100)GO TO 804
      CALL LOBASN(TIM,K,NSCD,RATE,ITD)
```

```
      IF(ITD.GE.100)GO TU 804
      CALL LSQQUA(TIM,ALPHA,NSMD,K,MID,SIA,SL,SNMB,SNMC,SM,ENX,EOD)
      SNMA=ALPHA+NSMD(MID)
      SNMT=TIM(MID)+.25
      SD=SIA
      WRITE(11,646)
  646 FORMAT(83H         ALPHA.NSM   MID.PT.   RATE             SIGMA
     X    BETA           GAMMA    )
      WRITE(11,152)ALPHA,MID,RATE,SIA,SNMB,SNMC
      K=JK
      CALL LSQQUA(TIM,ALPHA,NSCD,K,MID,SIA,SL,SNCB,SNCC,SM,ENE,EOE)
      SNCA=ALPHA+NSCD(MID)
      SNCT=TIM(MID)+.45
      SE=SIA
      RATE=(((SNMB*ANT/4.)+(SNCB*ANT/3.5))/2.)
      IF(ABS(RATE).LE..05)RATE=0.0
      WRITE(11,647)
  647 FORMAT(83H         ALPHA.NSC   MID.PT.   RATE             SIGMA
     X    BETA           GAMMA    )
      WRITE(11,152)ALPHA,MID,RATE,SIA,SNCB,SNCC
C     NSF LOBE ASSIGN.
      IF(ABS(SD).GT..1)GO TO 54
      IF(ABS(SE).GT..1)GO TO 56
   59 SL=.01
      SM=2.5
      K=JK
      CALL LOBASN(TIM,K,ENSF,RATE,ITD)
      IF(ITD.GE.100)GO TO 804
      CALL LSQQUA(TIM,ALPHA,ENSF,K,MID,SIA,SL,SNFB,SNFC,SM,ENF,EOF)
      MIDNS=MID
      SNFT=TIM(MID)
      SNFA=ALPHA+ENSF(MID)
  230 SF=SIA
      ISO=SIA*1000.+.5
  229 CALL ZERO(3,ISO,INNO)
      IL=0
      IK=0
      IJ=50.
      IF(ABS(SC).GT..05)GO TO 231
  234 IF(ABS(SF).GT..05)GO TO 235
      GO TO 239
  231 AST(6)=F1
      IK=50
      GO TO 234
```

```
  235 AST(6)=F2
      IL=50
  239 WRITE(11,645)
  645 FORMAT(83H         ALPHA.NSF   MID.PT.   RATE            SIGMA
     X    BETA             GAMMA    )
      WRITE(11,152)ALPHA,MID,RATE,SIA,SNFB,SNFC
      WRITE(11,153)SNMA,SNMT,SNCA,SNCT,RATE
      WRITE(11,777)
C     LOBE ASSIGNED DATA AFTER THE FIT HAS BEEN APPLIED **************
      WRITE(11,641)
  641 FORMAT(99H FRA.TIME  EWFINE   O-C   EWMEDM   O-C   EWCORS   O-C
     XNSFINE   O-C   NSMEDM   O-C   NSCORS   O-C )
      K=JK
      DO 790 I=1,K
      T=TIM(I)-EWMT-.15
      AM=EWMA+(EWMB*T)+(EWMC*T**2)
      T=TIM(I)-EWCT+.05
      AN=EWCA+(EWCB*T)+(EWCC*T**2)
      T=TIM(I)-EWFT
      AO=EWFA+(EWFB*T)+(EWFC*T**2)
      T=TIM(I)-SNMT+.25
      AP=SNMA+(SNMB*T)+(SNMC*T**2)
      T=TIM(I)-SNCT+.45
      AQ=SNCA+(SNCB*T)+(SNCC*T**2)
      T=TIM(I)-SNFT
      AR=SNFA+(SNFB*T)+(SNFC*T**2)
      AO=EEWF(I)-AO
      AM=EWMD(I)-AM
      AN=EWCD(I)-AN
      AR=ENSF(I)-AR
      AP=NSMD(I)-AP
      AQ=NSCD(I)-AQ
      WRITE(11,791)TIM(I),EEWF(I),AO,EWMD(I),AM,EWCD(I),AN,ENSF(I),AR,NS
     XMD(I),AP,NSCD(I),AQ
  791 FORMAT(X,F7.1,6(X,F7.3,X,F6.3))
  790 CONTINUE
      WRITE(11,792)ENA,SA,EOA,ENB,SB,EOB,ENC,SC,EOC
      WRITE(11,793)
      WRITE(11,794)
      WRITE(11,792)ENX,SD,EOD,ENE,SE,EOE,ENF,SF,EOF
  792 FORMAT(3(X,I2,4X,F5.3,8X,I2))
  793 FORMAT( 67H IN    EWM SIGMA    OT IN    EWC SIGMA    OT IN    EWF
     XSIGMA    OT )
  794 FORMAT( 67H IN    NSM SIGMA    OT IN    NSC SIGMA    OT IN    NSF
```

C.4

```
      XSIGMA     OT )                                                    0008340
C      ADJUST TIME TO EWFINE FITTED TIME                                 0008350
       XKFA=(KFA(L)/1000.)+IDIF                                          0008360
       XKFB=(KFB(L)/1000.)+IDIF                                          0008370
        EW4SEC=(EWFC*XKFA-EWFB)*XKFA                                     000838
        SN4SEC=(SNFC*(EWFT-SNFT-XKFB)+SNFB)*(EWFT-SNFT-XKFB)             000839
        EWFA=EWFA+EW4SEC                                                 000840
        SNFA=SNFA+SN4SEC                                                 000841
       EWFB=EWFB+2.*EWFC*(-XKFA)                                         0008420
       SNFB=SNFB+2.*SNFC*(-XKFB)                                         0008430
       EWMA=EWMA+EWMB*(EWFT-EWMT)+EWMC*((EWFT-EWMT)**2)                  0008440
       EWCA=EWCA+EWCB*(EWFT-EWCT)+EWCC*((EWFT-EWCT)**2)                  0008450
       SNMA=SNMA+SNMB*(EWFT-SNMT)+SNMC*((EWFT-SNMT)**2)                  0008460
       SNCA=SNCA+SNCB*(EWFT-SNCT)+SNCC*((EWFT-SNCT)**2)                  0008470
  118  EWFBB=EWFA-CEWF1                                                  0008480
C      PHASE ANGLE (FITTED) MINUS KS2+KC-KS1                             0008490
       EWMBB=EWMA-CEWM                                                   0008500
       EWCBB=EWCA-CEWC                                                   0008510
       SNFBB=SNFA-CNSF1                                                  0008520
       SNMBB=SNMA-CNSM                                                   0008530
       SNCBB=SNCA-CNSC                                                   0008540
C      REMOVE LOBE INTERGER                                              0008550
       LDIF=EWFBB                                                        0008560
       EWFBB=EWFBB-LDIF                                                  0008570
       LDIF=EWMBB                                                        0008580
       EWMBB=EWMBB-LDIF                                                  0008590
       LDIF=EWCBB                                                        0008600
       EWCBB=EWCBB-LDIF                                                  0008610
       LDIF=SNFBB                                                        0008620
       SNFBB=SNFBB-LDIF                                                  0008630
       LDIF=SNMBB                                                        0008640
       SNMBB=SNMBB-LDIF                                                  0008650
       LDIF=SNCBB                                                        0008660
       SNCBB=SNCBB-LDIF                                                  0008670
       AB=EWMBB-EWCBB                                                    0008680
       LDIF=AB                                                           0008690
       AB=AB-LDIF                                                        0008700
       CALL NORMAL(AB)                                                   0008710
       BC=SNMBB-SNCBB                                                    0008720
       LDIF=BC                                                           0008730
       BC=BC-LDIF                                                        0008740
       CALL NORMAL(BC)                                                   0008750
       AB4=AB*8.                                                         0008760
       BC4=BC*8.                                                         0008770
```

```
      AB3=AB*7.
      BC3=BC*7.
      EAB4=AB4-EWMBB
      LDIF=EAB4
      EAB4=EAB4-LDIF
      CALL NORMAL(EAB4)
      EBC4=BC4-SNMBB
      LDIF=EBC4
      EBC4=EBC4-LDIF
      CALL NORMAL(EBC4)
      DE=AB4-EAB4
      FG=BC4-EBC4
      EAB3=AB3-EWCBB
      LDIF=EAB3
      EAB3=EAB3-LDIF
      CALL NORMAL(EAB3)
      EBC3=BC3-SNCBB
      LDIF=EBC3
      EBC3=EBC3-LDIF
      CALL NORMAL(EBC3)
      DE3=AB3-EAB3
      FG3=BC3-EBC3
      CORE=DE+DE3
      CORN=FG+FG3
      COREF=(ANT/7.5)*CORE
      CORNF=(ANT/7.5)*CORN
      FE=COREF-EWFBB
      LDIF=FE
      FE=FE-LDIF
      CALL NORMAL(FE)
      FN=CORNF-SNFBB
      LDIF=FN
      FN=FN-LDIF
      CALL NORMAL(FN)
      EWLOBE=COREF-FE
      SNLOBE=CORNF-FN
C     AMBIGUITY ERRORS
      EWMER=4./ANT*EWLOBE-DE
      CALL NORMAL(EWMER)
      EWCER=3.5/ANT*EWLOBE-DE3
      CALL NORMAL(EWCER)
      SNMER=4./ANT*SNLOBE-FG
      CALL NORMAL(SNMER)
      SNCER=3.5/ANT*SNLOBE-FG3
```

```
      CALL NORMAL(SNCER)
      ABB=AB*2
      ADE=DE/4
      ADE3=DE3/3.5
      ACORE=CORE/7.5
      AEWLOB=EWLOBE/ANT
      WRITE(11,1152)ABB,ADE3,ADE,ACORE,AEWLOB,EWFT
      ABC=BC*2
      AFG=FG3/3.5
      AFG3=FG3/3.5
      AFG=FG/4
      ACORN=CORN/7.5
      ASNLOB=SNLOBE/ANT
      WRITE(11,1152)ABC,AFG3,AFG,ACORN,ASNLOB,SNFT
 1152 FORMAT(6(X,F14.6))
      IF(ANTD(K)-2)650,651,651
  651 WRITE(11,648)
  648 FORMAT(12H POLOR PASS )
      GO TO 652
  650 WRITE(11,649)
  649 FORMAT(17H EQUATORIAL PASS )
      GO TO 652
  652 IAO=2
      IBO=3
      SMITTY=((FREQ(M)/136.)*ANT)
      ACOS=EWLOBE/SMITTY
      BCOS=SNLOBE/SMITTY
      DCOS=     1.-ACOS*ACOS-BCOS*BCOS
      IF(DCOS.GE.0.0) DCOS = SQRT(DCOS)
      ADOT = EWFB/SMITTY
      BDOT = SNFB/SMITTY
      IF(DCOS.NE.0.) DDOT = -(ACOS * ADOT + BCOS * BDOT)/DCOS
      LL = L
      IF(ANT.EQ.57.) LL = L + 17
      IPASS(LL) = IPASS(LL) + 1
      WRITE(6,1017) ACOS,BCOS,DCOS,ADOT,BDUT,DDOT,IARC,IPASS(LL)
 1017 FORMAT(3X,4HL = ,F9.7,3X,4HM = ,F9.7,3X,4HN = ,F9.7,3X,7HLDOT = ,
     1F9.7,3X,7HMDOT = ,F9.7,3X,7HNDOT = ,F9.7/X,6HARC = ,I3,3X,
     17HPASS = ,I3)
      IF (LFILE) GO TO 1042
C     WRITE HEADINGS ON FILES
      DO 1040 LF=1,16
      LX=(LF-1)*2+20
      WRITE(LX,1051) STATIO(LF),KSTA(LF)
```

```
      WRITE (LX,1052)
      LX=LX+1
      WRITE (LX ,1051) STATIO(LF),KSTA(LF+17)
      WRITE (LX ,1052)
 1040 CONTINUE
      LFILE=.TRUE.
C     DETERMINE FILE NUMBER
 1042 LF=(L-1)*2 +20
      IF(ANT .EQ. 57.) LF=LF+1
 1050 FORMAT (1X,I3,3(1X,I3)    ,2X,I3,2X,I3,2X,6(2X,F9.6),2(3X,F9.6))
 1051 FORMAT (// 1X,    '++++++++++++++++++++++++++++++++++++++++++++
     *++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
     *+++++++'  //
     *          20X,'STATION NAME  ',A6/20X,'STATION NO.   ',I4/)
 1052 FORMAT (1X,'DAY',5X,'GMT',5X,'ARC',2X,'PASS',7X,'L',10X,'M'10X,
     1          'N',8X,'LDOT'8X,'MDOT', 8X,'NDOT',
     1          3X,'KC-KS1+KS2',3X,'KC-KS1+KS2'/
     2          6X,'HR MIN SEC',83X,'(EW)',8X,'(NS)'//)
      IGMH=TIM(MID)/3600.
      IGMM=(TIM(MID)-(IGMH*3600.))/60.
      IGMS=TIM(MID)-IGMH*3600.-IGMM*60.
      WRITE(LF,1050) IDAYD(MID),IGMH,IGMM,IGMS,IARC,IPASS(LL),
     *           ACOS,BCOS,DCOS,ADOT,BDOT,DDOT,CEWF1,CNSF1
      IF(SMITTY.GT.0.D0) GO TO 325
      IF((ABS(ACOS).LT.CCOS).AND.(ABS(BCOS).LT.CCOS))GO TO 325
      WRITE(13,341)IB1,CSTA,STATIO(L)
      WRITE(13,326)KDEG
  326 FORMAT(1H+,26X,'ELEV. LESS THAN ',I2,' DEG.')
      GO TO 94
  325 CONTINUE
       EWLOBE = EWLOBE - EW4SEC
       SNLOBE = SNLOBE - SN4SEC
       XKFA = 0.4 - XKFA
       XKFB = 0.4 - XKFB
      EWTEMP=EEWF(MIDEW)-CEWF1
      SNTEMP=ENSF(MIDEW)-CNSF1
      CALL FIXUP1(EWLOBE,EWTEMP,JOE)
      CALL FIXUP1(SNLOBE,SNTEMP,JON)
      DO 301 IFOV=1,NBRK
      EEWF(IFOV)=EEWF(IFOV)+JCE-CEWF1
      ENSF(IFOV)=ENSF(IFOV)+JCN-CNSF1
      DO 327 IFIX=1,5
      TFOVEW(IFOV,IFIX)=(TFOVEW(IFOV,IFIX)+TFOVNS(IFOV,IFIX)-XKFA
     *                 -XKFB)*0.5
```

```
      DELTAT=TFOVEW(IFOV,IFIX)-TFOVNS(IFOV,IFIX)+XKFB
      FOVEW(IFOV,IFIX)=FOVEW(IFOV,IFIX)-DELTAT * EWFB -CEWF1
      FOVNS(IFOV,IFIX)=FOVNS(IFOV,IFIX)+DELTAT *SNFB - CNSF1
  327 CONTINUE
      CALL FIXUP2(FOVEW(IFOV,3),EEWF(IFOV))
      CALL FIXUP2(FOVNS(IFOV,3),ENSF(IFOV))
      CALL FIXUP2(FOVEW(IFOV,2),FOVEW(IFOV,3))
      CALL FIXUP2(FOVEW(IFOV,1),FOVEW(IFOV,2))
      CALL FIXUP2(FOVEW(IFOV,4),FOVEW(IFOV,3))
      CALL FIXUP2(FOVEW(IFOV,5),FOVEW(IFOV,4))
      CALL FIXUP2(FOVNS(IFOV,2),FOVNS(IFOV,3))
      CALL FIXUP2(FOVNS(IFOV,1),FOVNS(IFOV,2))
      CALL FIXUP2(FOVNS(IFOV,4),FOVNS(IFOV,3))
      CALL FIXUP2(FOVNS(IFOV,5),FOVNS(IFOV,4))
      DO 301 JOUT=1,5
      AACOS(IFOV,JOUT)=FOVEW(IFOV,JOUT)/SMITTY
      BBCOS(IFOV,JOUT)=FOVNS(IFOV,JOUT)/SMITTY
  301 CONTINUE
      N5=5*NBRK
      IPAGE=1
      IGMT=TIM(1)
      WRITE(6,310)IPAGE,IB1,IGMT
      IPAGE=0
      WRITE(6,302)IPAGE,(MSGR(I),I=1,80)
      IPAGE=1
      WRITE(6,306)KKSAID,STATIO(L),IDAYD(1),N5
  306 FORMAT('0SATELLITE = ',I6,7X,'STATION = ',A6,6X,'DAY = 'I3,7X,
     1'DATA CARDS = ',I3/)
      WRITE(6,308)
  308 FORMAT(6X,'EWF TIME',2X,'EWF COUNT',6X,'L',9X,'NSF TIME',2X,'NSF C
     1OUNT',6X,'M'/)
  307 FORMAT(' SATELLITE = ',I6,7X,'STATION = ',A6,6X,'DAY = 'I3,7X,
     1'DATA CARDS = ',I3)
  309 FORMAT(6X,'EWF TIME',2X,'EWF COUNT',6X,'L',9X,'NSF TIME',2X,'NSF C
     1OUNT',6X,'M')
  310 FORMAT(I1,'&',/,' DATA MESSAGE NO. ',I5,6X,'GMT IN SEC. ',I5/)
  311 FORMAT(  ' &',/,' DATA MESSAGE NO. ',I5,6X,'GMT IN SEC ',I5)
      DO 303 IOUT=1,NBRK
      DO 303 JOUT=1,5
      KLM=(IOUT-1)*5+JOUT
  303 WRITE(6,304)TFOVEW(IOUT,JOUT),FOVEW(IOUT,JOUT),AACOS(IOUT,JOUT),
     *  TFOVEW(IOUT,JOUT),FOVNS(IOUT,JOUT),BBCOS(IOUT,JOUT),KLM
  304 FORMAT(2(4X,F10.5,2X,F9.5,2X,F8.5),6X,I4)
      JDY=JULDY(IYEAR,IDAYD(1))
```

```
      L = LL                                                            001054
      IF(JDY.LT.JUDY(L))GO TO 331                                       0010550
      IF((JDY.EQ.JUDY(L)).AND.(IGMT.LT.JUSEC(L)))GO TO 331              0010560
      JUSEC(L)=IGMT                                                     0010570
      JUDY(L)=JDY                                                       0010580
      DNAP(1)=0.D0                                                      0010590
       DNAP(2) = IARC                                                   001060
      DNAP(3)=KSTA(L)                                                   0010610
      DNAP(4)=IPASS(L)                                                  0010620
      WRITE(12)(DNAP(I),I=1,6)                                          C010630
      WRITE (19) DNAP                                                   001064
      JDAY=JDY                                                          0010650
      DNAP(1)=JDAY                                                      0010660
      JDAY1=1                                                           0010670
      I=2*N5                                                            0010680
       WRITE(14,353)IB1,I,KSTA(L),STATIO(L),IARC,IPASS(L),JDAY,IGMT     001069
     *     ,ACOS,BCOS,DCOS,ADCT,BDOT,DDOT                               001070
  353 FORMAT (3X,I5,I6,2X,I3,2X,A6,I4,I5,2X,I5,I6,6(2X,F8.6))           001071
      DO 332 IOUT=1,NBRK                                                0010720
      DO 332 JOUT=1,5                                                   0010730
      DNAP(2)=IGMT+TFOVEW(IOUT,JOUT)                                    0010740
      DNAP(3)=AACOS(IOUT,JOUT)                                          0010750
      DNAP(4)=BBCOS(IOUT,JOUT)                                          0010760
  335 IF(DNAP(2).LT.86400.D0)GO TO 333                                  0010770
      JDAY=JDAY+JDAY1                                                   0010780
      DNAP(1)=JDAY                                                      0010790
      JDAY1=0                                                           0010800
      DNAP(2)=DNAP(2)-86400.D0                                          0010810
  333 WRITE(12)(DNAP(I),I=1,6)                                          0010820
  332 CONTINUE                                                          0010830
      I=DNAP(2)                                                         0010840
      XNAP(1)=JULDY(IYEAR,IDAYD(MID))                                   001085
      XNAP(2)=TIM(MID)                                                  001086
      XNAP(3)=ACOS                                                      001087
      XNAP(4)=BCOS                                                      001088
      WRITE(19) XNAP                                                    001089
       IF(IARCS.LT.0) GO TO 1995                                        001090
       II = 1                                                           001091
       IF(IARCS.NE.IARC) II = 2                                         001092
       WRITE(7,2501)  STATIO(LS),POLEQ,IARCS,JS,IPASSS,II               001093
 1995  IF(ANT.EQ.57.) GO TO 2C00                                        001094
       POLEQ = EQ                                                       001095
       JS = L + L - 1                                                   001096
       GO TO 2100                                                       001097
```

```
 2000 POLEQ = POL
      JS = L + L - 34
 2100 TIMSTR = IGMT + TFOVEW(1,1) - 1.
      TIMEND = IGMT + TFOVEW(NBRK,5) + 1.
      WRITE(7,2500) I201,STATIO(L),POLEQ,IARC,JS,IPASS(L),
     * IYEAR,IDAYD(1),TIMSTR
      WRITE(7,2500) I202,STATIO(L),POLEQ,IARC,JS,IPASS(L),
     * IYEAR,IDAYD(1),TIMEND
      IARCS = IARC
      LS = L
      IPASSS = IPASS(L)
 2500 FORMAT(I3,2X,A6,A2,3I3,6X,I3,'  1',I3,10X,F9.1)
 2501  FORMAT ('999',2X,A6,A2,4I3)
     GO TO 94
  331 CONTINUE
     WRITE(13,341)IB1,CSTA,STATIO(L)
  341 FORMAT(' DATA MESSAGE NO.',I5,38X,'STATION NO.',I3,6X,'STATION NAM
    1E ',A6)
     WRITE(13,342)
  342 FORMAT(1H+,26X,'MESSAGE OUT OF SEQUENCE')
     GO TO 94
   30 WRITE( 6,31)
   31 FORMAT(17H CAL.LINE PARITY )
     WRITE(13,341)IB1,CSTA,STATIO(L)
     WRITE(13,343)
  343 FORMAT(1H+,33X,'CAL. LINE PARITY')
     GO TO 94
   32 WRITE(11,33)
   33 FORMAT(24H DATA PARITY,NO MSG.END )
     GO TO 621
  604 CONTINUE
     WRITE(13,341)IB1,CSTA,STATIO(L)
     WRITE(13,344)
  344 FORMAT(1H+,34X,'CAL. LINE ERROR')
     DO 60 JZ=1,60
     JY=JZ
     IF(JY.EQ.32)GO TO 94
   61 READ(9,601,END=902,ERR=61)(DATA(I),I=1,65)
     IF(DATA(1).EQ.IAMP)GO TO 94
     IF(DATA(10).EQ.SPX)GO TO 62
     GO TO 64
   62 IF(DATA(30).EQ.SPX)GO TO 63
     GO TO 64
   63 IF(DATA(50).EQ.SPX)GO TO 39
```

```
   64 WRITE( 6,660)(DATA(I),I=1,65)
  660 FORMAT(X,65A1)
   60 CONTINUE
      GO TO 94
   65 K=JK-1
      DO 66 I=1,K
      WRITE( 6,67)HORD(I),MIND(I),SECD(I),EEWF(I),EWMD(I),EWCD(I),ENSF(I
     X),NSMD(I),NSCD(I),AMTD(I),IDAYD(I),STAD(I)
   67 FORMAT(X,I2,I2,I2,X,(6(F10.6,X)),I2,X,I3,X,I2,X)
   66 CONTINUE
      GO TO 94
  777 FORMAT(1H1)
    1  CONTINUE
       DNAP(1) = -1.D0
       DNAP(2) = 0.D0
       DNAP(4) = 0.D0
       WRITE(12) (DNAP(I), I = 1,6)
      WRITE(19) DNAP
      END FILE 19
      REWIND 19
    3  WRITE (6,990)
      WRITE(11,990)
      WRITE(13,340)(MSGR(I),I=1,80)
      WRITE(13,349)
  349 FORMAT(3X,'SUMMARY OF PASSES'//3X,'STATION NO.',9X,'STATION NAME',
     113X,'PASSES',14X,'JUL DAY',17X,'SEC')
       II = 3
       WRITE(7,2501)  STATIO(LS),POLEQ,IARCS,JS,IPASSS,II
      DO 350 J=1,17
      I=J
      DO 350 JJ=1,2
      IPASST(I) = IPASST(I) + IPASS(I)
       WRITE(13,351)KSTA(I),STATIO(I),IPASST(I),JUDY(I),JUSEC(I)
      I=I+17
  350 CONTINUE
  351 FORMAT(4X,I5,17X,A6,15X,I5,2(17X,I5))
      WRITE(13,990)
      END FILE 12
      REWIND 12
  990 FORMAT(50X,'JOB IS COMPLETE',2(/1H1))
      DO 355 J=1,2
      N1=17*J
      N=N1-16
      WRITE(13,336)A1,N,(IPASS(I),I=N,N1)
```

```
  355 CONTINUE
      DO 356 J=1,2
      N1=17*J
      N=N1-16
      WRITE(13,336)A2,N,(JUDY(I),I=N,N1)
  356 CONTINUE
      DO 357 J=1,2
      N1=17*J
      N=N1-16
      WRITE(13,336)A3,N,(JUSEC(I),I=N,N1)
  357 CONTINUE
  336 FORMAT(1X,A5,18I5)
      STOP
  321 WRITE(6,322)
  322 FORMAT('1 *STOP* NO DATA IN FT08 FILE')
      STOP
      END
      SUBROUTINE FIXUP1(EWLOBE,EEWF,JOE)
      FOVEWF=EWLOBE-EEWF
      JOE=FOVEWF
      IF(ABS(FOVEWF-JOE)-0.5)1,1,2
    2 IF(FOVEWF-JOE)3,3,4
    3 JOE=JOE-1
      RETURN
    4 JOE=JOE+1
    1 RETURN
      END
      SUBROUTINE FIXUP2(EWMID,EEWF)
      IF(EEWF.LT.0.0) EWMID=EWMID-1.0
      DIFF=EEWF-EWMID
      NDIFF=DIFF
      IF(ABS(DIFF-NDIFF)-0.5)1,1,2
    1 EWMID=EWMID+NDIFF
      RETURN
    2 IF(DIFF)3,3,4
    3 NDIFF=NDIFF-1
      GO TO 1
    4 NDIFF=NDIFF+1
      GO TO 1
      END
      SUBROUTINE LOBASN(TEM,I,A,RATE,L)
      DIMENSION A(31),TEM(31)
      K=I-1
      IF(RATE.GT.0.)A(1)=A(1)+1.
```

```
      IF(RATE.LT.0.)A(1)=A(1)-1.
      DO 10 J=1,K
      L=0
      JJ=J+1
      DELTA=RATE*(TEM(JJ)-TEM(J))
   11 X=A(JJ)-A(J)
      L=L+1
      IF(L.GE.100)GO TO 12
      IF(ABS(DELTA-X)-.500)10,10,6
    6 IF(DELTA)4,5,3
    4 A(JJ)=A(JJ)-1
      GO TO 11
    3 A(JJ)=A(JJ)+1
      GO TO 11
    5 IF(X)3,10,4
   10 CONTINUE
   12 RETURN
      END
      SUBROUTINE LSQQUA(T,ALPHA,XY,ITO,L,SIGMA,F,BETA,GAMMA,SM,INN,IOT)
      DIMENSION B(3,4),A(3,4),T(32),XY(32)
      DO 1 I=1,3
      DO 1 J=1,4
    1 A(I,J)=0.0
      L=(ITO+1)/2
      A(1,1)=ITO
      INN=ITO
      DELSQ=0.
      DO 10 I=1,ITO
      RAPPA=XY(I)-XY(L)
      TAU=T(I)-T(L)
      A(1,2)=A(1,2)+TAU
      A(1,3)=A(1,3)+TAU**2
      A(1,4)=A(1,4)+RAPPA
      A(2,3) =A(2,3)+TAU**3
      A(2,4)=A(2,4)+RAPPA*TAU
      A(3,3) =A(3,3) +TAU**4
      A(3,4)=A(3,4)+RAPPA*TAU**2
   10 DELSQ=RAPPA**2+DELSQ
   67 A(2,2)=A(1,3)
      A(2,1)=A(2,4)
      A(3,1)=A(3,4)
      N=1
      DO20 K=1,3
      N=N+1
```

```
      DO15 J=N,4
   15 B(K,J)=A(K,J)/A(K,K)
      IF(N-4)12,21,11
   12 DO20I=N,3
      DO20J=I,4
   20 A(I,J)=A(I,J)-A(K,I)*B(K,J)
   21 GAMMA=B(3,4)
      BETA=B(2,4)-GAMMA*B(2,3)
      ALPHA    =B(1,4)-BETA*B(1,2)-GAMMA*B(1,3)
      N=A(1,1)
      SIGMA=((DELSQ-ALPHA*A(1,4)-BETA*A(2,1)-GAMMA*A(3,1))/A(1,1))
      IF(SIGMA)70,71,71
   70 SIGMA=0.
   71 SIGMA=SQRT(SIGMA)
      IF(SIGMA-F   )59,59,35
   35 DO61I=1,3
      DO61J=1,4
   61 A(I,J)=0.0
      DELSQ=0.0
      DO65I=1,ITO
      RAPPA=XY(I)-XY(L)
      TAU= T(I)-T(L)
      R=RAPPA-ALPHA-BETA*TAU-GAMMA*TAU**2
      IF(R)2,3,3
    2 R=-R
    3 IF(R-SM*SIGMA)32,32,65
   32 A(1,1)=A(1,1)+1.
      A(1,2)=A(1,2)+TAU
      A(1,3)=A(1,3)+TAU**2
      A(1,4)=A(1,4)+RAPPA
      A(2,3) =A(2,3)+TAU**3
      A(2,4)=A(2,4)+RAPPA*TAU
      A(3,3) =A(3,3) +TAU**4
      A(3,4)=A(3,4)+RAPPA*TAU**2
      DELSQ=RAPPA**2+DELSQ
   65 CONTINUE
      IF(A(1,1)-5)59,66,66
   66 IF(A(1,1)-N   )67,59,67
   59 ITO=A(1,1)
      IOT=ITO
      RETURN
   11 WRITE( 6,7)
    7 FORMAT(30H THIS MSG. HAS EXCESSIVE NOISE)
      RETURN
```

```
      END
      SUBROUTINE NORMAL(X)
      IF(X)113,114,115
  113 IF(ABS (X)-.5)114,114,116
  116 X=X+1.0
      GO TO 114
  115 IF(ABS (X)-.5)114,114,117
  117 X=X-1.0
  114 RETURN
      END
      SUBROUTINE ZERO(N,IIN,AREA)
      DIMENSION DIV(7)
      INTEGER DIV
      LOGICAL*1 PLUS,MINUS,ASK,IC(10),AREA(80)
      DATA PLUS,MINUS,ASK,IC(1),IC(2),IC(3),IC(4),IC(5),IC(6),IC(7),IC(8
     X),IC(9),IC(10)/Z40,Z60,Z5C,ZF0,ZF1,ZF2,ZF3,ZF4,ZF5,ZF6,ZF7,ZF8,ZF9
     X/
      DATA DIV(1),DIV(2),DIV(3),DIV(4),DIV(5),DIV(6),DIV(7)/1000000,1000
     X00,10000,1000,100,10,1/
      AREA(1)=PLUS
      IF(IIN)1,2,2
    1 AREA(1)=MINUS
      IIN=IABS(IIN)
    2 IJ=9-N
      K=2
      J=N-1
      DO 4 I=1,J
      ITEMP=IIN/DIV(IJ)
      IF(ITEMP.LE.9.)GO TO 6
      AREA(K)=ASK
      GO TO 5
    6 AREA(K)=IC(ITEMP+1)
    5 IIN=IIN-(ITEMP*DIV(IJ))
      IJ=IJ+1
      K=K+1
    4 CONTINUE
      RETURN
      END
      INTEGER FUNCTION JULDY(Y,D)
      INTEGER Y,D
      L=(Y-50+1)/4
      JULDY=L*366
      JULDY=(Y-50-L)*365+JULDY+D-1
      RETURN
```

```
END                                                   0013620

                                                                 1371 CARDS
```

A-2.4 A PROGRAM FOR MERGING TWO OPTICAL DATA TAPES (IN A GEOS FORMAT) INTO A SINGLE TAPE - SAMPLE JCL

```
C
C
C
C     PROGRAM FOR COMBINING TWO GEOS-FCRMATED DATA TAPES
C     FOR USE BY OPTICAL PREPROCESSOR PROGRAM
C
C
      REAL*8 TAPE
      INTEGER*2 ID,I10
      LOGICAL*1 A,B,AA,BB,ZERC,BLANK,AA1,BB1,I9
      DIMENSION A(78),B(78),AA(2),BB(2)
      EQUIVALENCE (A(17),IYMA),(A(21),IDHA),(A(25),IMSA),(A(29),IFSA),
     1(B(5),ID),(B(17),IYMB),(B(21),IDHB),(B(25),IMSB),(B(29),IFSB),
     2(AA1,AA(1)),(BB1,BB(1))
      COMMCN TAPE,IFILE,IYM,INIT,LINE
      DATA ZERO,BLANK/'0',' '/
      DATA I9,I10/'9','10'/
      INIT = 0
      READ (5,900) TAPE,IFILE
      READ (1,100C,END = 300) AA,A
      READ (1,1000,END = 300) AA,A
      READ (1,1000,END = 300) AA,A
      DO   5 I = 23,32
      IF (A(I).EQ.BLANK) A(I) = ZERO
    5 CONTINUE
   10 READ (2,1000,END = 40C) BB,B
      IF (ID.NE.I10) GO TC 10
      DO  15 I = 23,32
      IF (B(I).EQ.BLANK) B(I) = ZERO
   15 CONTINUE
C
      IF (IYMA - IYMB) 20,20,30
   20 CALL RESET (IYMA)
      GO TO 35
   30 CALL RESET (IYMB)
   35 INIT = -1
   40 CONTINUE
      IF (IYMA - IYMB) 100,50,200
   50 IF (IDHA - IDHB) 10C,60,2C0
   60 IF (IMSA - IMSB) 10C,70,2C0
   70 IF (IFSA.GT.IFSB) GC TO 2C0
  100 CONTINUE
      IF (IYMA.GT.IYM) CALL RESET (IYMA)
      WRITE (3,1000) AA,A
```

```
      IF(KEY7L.GT.0) PLOT(1)=STAR
      IF(KEY7M.GT.0) PLUT(2)=STAR
      WRITE(LF,1107) IDAY,IH,IM,SEC,NOARC,NOPASS,LERROR,MERROR,ELEV,WT,
     *LSD,MSD,LLOBE,MLOBE,NOBS,NEDIT,SLOPE,IROPE,DMS,PLOT
      GO TO 10
  200 RETURN
  500 ERR(I)=10.D50
      KI(I)=1
      GO TO 30
  520 IF (ERRORP.GT.20.D0) KI(I)= IGRAPH(I)+20
      IF(ERRORP.LT.-20.D0) KI(I)= IGRAPH(I)-20
      GRAPH(KI(I))= STAR
      GO TO 30
 1104 FORMAT (//1X,132('=')//10X,A8,10X,'STATION ID',I4//47X,'IN      STD
     *     LOBE    NUMBER   PTS',6X,'ERROR RATE',7X,'RD IN     COMPUTED
     *     NEW'/32X'ERROR    ELEVA SOL    DEV     CORR    OF PTS  DELTD'
     *,22X,'ST DV    MEASUREMENT    WT'/1X,'DAY HR MN SECOND ARC PASS
     * L     M   TION L M   L    M    L    M    L    M    L  M      L',8X,'
     *M',7X,'L  M      L',8X,'M     L M'//)
 1105 FORMAT(///1X,132('=')///1X,A8,' STATION',I4,' ARC',I3,' PASS',
     1I3/43X,'MIDPOINT TIME',14X,'MEAN  ERROR',9X,'STANDARD DEVIATION'/
     242X,'DAY HR MN SECOND',2(12X,'L',9X,'M')/2X,
     3'TIME        ERROR RELATIVE TO',12X,3I3,F7.3,6X,2F10.5,4X,2F9.5/
     41X,'RELATIVE    MEAN PASS ERROR'/1X,'TO MIDPT          L',9X,'M'/)
 1106 FORMAT(F9.3,2F10.5,5X,83A1)
 1107 FORMAT (1X,3I3,F7.3,I3,I5,1X,2I6,5X,A1,3X,A1,1X,A1,2I4,2(1X,2I4),1
     *X,2I3,1X,2D9.2,1X,2I3,2F9.5,2(1X,A1))
      END
```

0424 CARDS

```
C
C
C
C     PROGRAM FOR COMBINING TWO GEOS-FCRMATED DATA TAPES
C     FOR USE BY OPTICAL PREPROCESSOR PROGRAM
C
C
      REAL*8 TAPE
      INTEGER*2 ID,I10
      LOGICAL*1 A,B,AA,BB,ZERC,BLANK,AA1,BB1,I9
      DIMENSION A(78),B(78),AA(2),BB(2)
      EQUIVALENCE (A(17),IYMA),(A(21),IDHA),(A(25),IMSA),(A(29),IFSA),
     1(B(5),ID),(B(17),IYMB),(B(21),IDHB),(B(25),IMSB),(B(29),IFSB),
     2(AA1,AA(1)),(BB1,BB(1))
      COMMGN TAPE,IFILE,IYM,INIT,LINE
      DATA ZERO,BLANK/'0',' '/
      DATA I9,I10/'9','10'/
      INIT = 0
      READ (5,900) TAPE,IFILE
      READ (1,100C,END = 300) AA,A
      READ (1,1000,END = 300) AA,A
      READ (1,1000,END = 300) AA,A
      DO   5 I = 23,32
      IF (A(I).EQ.BLANK) A(I) = ZERO
    5 CONTINUE
   10 READ (2,1000,END = 40C) BB,B
      IF (ID.NE.I10) GO TC 10
      DO  15 I = 23,32
      IF (B(I).EQ.BLANK) B(I) = ZERO
   15 CONTINUE
C
      IF (IYMA - IYMB) 20,20,30
   20 CALL RESET (IYMA)
      GO TO 35
   30 CALL RESET (IYMB)
   35 INIT = -1
   40 CONTINUE
      IF (IYMA - IYMB) 100,50,200
   50 IF (IDHA - IDHB) 10C,60,200
   60 IF (IMSA - IMSB) 10C,70,2C0
   70 IF (IFSA.GT.IFSB) GC TO 200
  100 CONTINUE
      IF (IYMA.GT.IYM) CALL RESET (IYMA)
      WRITE (3,1000) AA,A
```

```
      WRITE (6,1010) AA,A
      LINE = LINE - 1
      IF (LINE.LT.0) CALL TITLE
      READ (1,1000,END = 300) AA,A
      IF(AA1.EQ.I9) GO TO 300
      DO 150 I = 23,32
      IF (A(I).EQ.BLANK) A(I) = ZERO
  150 CONTINUE
      GO TO 40
  200 IF ('YMB.GT.IYM) CALL RESET (IYMB)
      WRITE (3,1000) BB,B
      WRITE (6,1010) BB,B
      LINE = LINE - 1
      IF (LINE.LT.0) CALL TITLE
  210 READ (2,1000,END = 400) BB,B
      IF(BB1.EQ.I9) GO TO 400
      IF (ID.NE.I10) GO TO 210
      DO 215 I = 23,32
      IF (B(I).EQ.BLANK) B(I) = ZERO
  215 CONTINUE
      GO TO 40
  300 IF (IYMB.GT.IYM) CALL RESET (IYMB)
      WRITE (3,1000) BB,B
      WRITE (6,1010) BB,B
      LINE = LINE - 1
      IF (LINE.LT.0) CALL TITLE
  310 READ (2,1000,END = 500) BB,B
      IF(BB1.EQ.I9) GO TO 500
      IF (ID.NE.I10) GO TO 310
      GO TO 300
  400 IF (IYMA.GT.IYM) CALL RESET (IYMA)
      WRITE (3,1000) AA,A
      WRITE (6,1010) AA,A
      LINE = LINE - 1
      IF (LINE.LT.0) CALL TITLE
      READ (1,1000,END = 500) AA,A
      IF(AA1.EQ.I9) GO TO 500
      GO TO 400
  500 CONTINUE
      END FILE 3
      REWIND 1
      REWIND 2
      REWIND 3
      WRITE (6,1040)
```

```
      STOP
  900 FORMAT (A8,2X,I5)
 1C00 FORMAT (80A1)
 1010 FORMAT (6X,6A1,3X,2A1,2X,5A1,3X,5A1,3X,6A1,3X,2A1,':',2A1,':',2A1,
     1'.',4A1,3X,3A1,':',2A1,':',2A1,'.',3A1,3X,3A1,':',2A1,':',2A1,'.',
     12A1,3X,27A1)
 1040 FORMAT (//'JOB COMPLETEC')
      END
      SUBROUTINE RESET (IYMC)
      REAL*8 TAPE
      COMMCN TAPE,IFILE,IYM,INIT,LINE
      IYM = IYMC
      IF (INIT.GE.0) GO TC 10
      IFILE = IFILE + 1
      END FILE 3
   10 CONTINUE
      CALL TITLE
      RETURN
      END
      SUBROUTINE TITLE
      REAL*8 TAPE
      INTEGER*2 IY
      DIMENSION IY(2)
      EQUIVALENCE (IYM,IY(1))
      COMMCN TAPE,IFILE,IYM,INIT,LINE
      WRITE (6,10C0) TAPE,IFILE,IY
      LINE = 55
      RETURN
 1C00 FORMAT ('1'/16X,'TAPE',2X,A8,'  FILE',I3,10X,'YEAR  ',A2,'  MONTH
     * ',A2//6X,'SATNO',3X,'TYPE',9X,'STAT',5X,'DATE',9X,'TIME',6X,'RIGH
     *T ASCENSION',3X,'DECLINATION'/)
      END

/*
//  EXEC  LINKGO
//GO.FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
//        SPACE=(CYL,(30,3))
//GO.FT01FC01 DD UNIT=240C-9,LABEL=(1,BLP),DISP=(OLD,DELETE),
//  DCB=(RECFM=FB,LRECL=8C,BLKSIZE=3200),VOL=SER=1664J
//GO.FT02F001 DD UNIT=2400-9,LABEL=(1,BLP),DISP=(OLD,DELETE),
```

```
//   DCB=(RECFM=FB,LRECL=8C,BLKSIZE=3200),VOL=SER=1666H
//GO.FT03F001 DD UNIT=2400-9,LABEL=(12,BLP),DISP=(NEW,PASS),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200,DEN=2),VOL=SER=3826F
//GO.FT03FC02 DD UNIT=240C-9,LABEL=(13,BLP),DISP=(NEW,DELETE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200,DEN=2),VOL=SER=3826F
//GO.DATA5 DD *
3826F         12
/*
```

0140 CARDS

A-2.5 OPTICAL PREPROCESSING PROGRAM - SAMPLE JCL

```
C
C
C
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
C
      DIMENSION A(13),NOSTA(3C),NONAME(30),C(80),MONTH(12)
C
      COMMON/W/ TIM(49,30),RA(49,30),DEC(49,30),STNAM(30),TPASS,
     1JDAYA(30),NUM(30),NCRDER(30),IYR1,IYR2,IMO1,IMO2,IDA1,IDA2,JDAY1,
     1JDAY2,IARC,IARCS,ISTA(3C),JSTA(30),NPASS(30),KSTA(30),KEY4,K,IS
      COMMCN /M/ TIM2
C
C
C
C     THE NUMBERS IN ARRAY MONTH ARE THE DAY NUMBERS OF THE LAST DAY OF
C     THE PRECEDING MONTH.  THE DAY NUMBERS ARE OBTAINED BY NUMBERING
C     THE CAYS OF THE YEAR CONSECUTIVELY STARTING WITH DAY NUMBER ZERO
C     ON 1 JANUARY.
C
      DATA MONTH/-1,30,58,89,119,150,180,211,242,272,303,333/
      DATA MINUS/'-'/
C
C
C     TORBT   IS USED BY THE PRCGRAM TC SEPARATE ORBITS. OBSERVATIONS
C     (FOR ANY ONE STATION) SEPARATED BY MORE THAN TORBT SECONDS ARE
C     REGARDED AS BELONGING TC CIFFERENT ORBITS.
C     TPASS   IS USED BY THE PROGRAM TO SEPARATE PASSES.  A PASS IS
C     REGARDED AS CONSISTING CF THE OBSERVATIONS OF ONE PHOTOGRAPHIC
C     PLATE.  OBSERVATIONS(FOR ANY ONE STATION) SEPARATED BY MORE THAN
C     TPASS SECONDS ARE REGARCED AS BELONGING TO DIFFERENT PASSES.
C     DELT  IS THE TIME(IN SECONDS) ADDED TO THE OBSERVATION TIME TO
C     ACCCUNT FOR SATELLITE FLASH BUILC-UP.
C     NMAX    IS THE MAXIMUM NLMBER OF OBSERVATIONS THAT THE PROGRAM
C     CAN ACCEPT FOR ANY CNE STATION AND ORBIT.
C
      TORBT = 2700.D0
      TPASS = 45.D0
      DELT = .0005D0
```

```
      NMAX = 49
C
      REWIND 1
      READ (1,1080) C
      READ (1,1080) C
      REWIND 2
C
C     SET UP INITIAL CONDITIONS.  SETTING KEY4 TO A NEGATIVE NUMBER
C     ENSURES THAT THE FIRST NAP CARD PUNCHED BY SUBROUTINE DATAWT
C     IS A '201' CARD.
C     'IS'IS THE NUMBER OF STATIONS FOR WHICH OBSERVATIONS HAVE BEEN
C     ACCUMULATED BUT NOT YET OUTPUT ON TAPE.
C     IYMD2  IS THE CURRENT (YEAR,MONTH,DAY).  SETTING IT TO A LARGE
C     NUMBER ENSURES THAT WHEN THE FIRST OBSERVATION PAST THE START TIME
C     FOR THE ARC IS READ IN,THE PROGRAM WILL ASSUME THAT A NEW DAY HAS
C     COMMENCED.
C     IYMD IS, IN GENERAL, THE (YEAR,MONTH,DAY) OF THE LAST OBSERVATION
C     READ IN.  HOWEVER, THE FIRST TIME IT IS USED NOTHING HAS BEEN
C     READ IN.      SETTING IT TO A LARGE NEGATIVE NUMBER ENSURES THAT
C     THE PROGRAM WILL CONTINUE TESTING FOR THE START OF THE FIRST ARC.
C     JDAY2 IS THE CURRENT JULIAN DAY NUMBER. SETTING IT EQUAL TO A
C     LARGE NEGATIVE NUMBER ENSURES THAT THE PEVIOUS JULIAN DAY NUMBER
C     ALWAYS IS LESS THAN THE CURRENT ONE, SINCE THE NEXT OBSERVATION
C     READ IN MUST OF NECESSITY HAVE A LARGER JULIAN DAY NUMBER.
C     TIM2 IS SET TO A LARGE NEGATIVE NUMBER (-1.D+30) TO INITIALIZE
C     SUBROUTINE TIMCOR.
C
      KEY4 = -1
      IS = 0
      IYMD2 = 1000000
      IYMD = -1000000
      JDAY2 =-1000000
      IARC = -1000000
      TIM2 =-1.D+30
      NOLIST = 0
C
C     SECATR IS THE CONVERSION CONSTANT FROM ARC-SECONDS TO RADIANS.
C     SECTTR IS THE CONVERSION CONSTANT FROM HOUR-ANGLE SECONDS TO
C     RADIANS.
C
      SECATR = .0174532925200/3600.D0
      SECTTR = SECATR*15.D0
      WRITE(3,1060)
      DO 10 NUMSTA = 1,30
```

```
      READ(5,1000)  KSTA(NUMSTA),STNAM(NUMSTA),ISTA(NUMSTA),IEND
      WRITE(3,1001) KSTA(NUMSTA),STNAM(NUMSTA),ISTA(NUMSTA)
      JSTA(NUMSTA) = ISTA(NUMSTA) - (ISTA(NUMSTA)/10000) * 10000
      NUM(NUMSTA) = 0
      IF (IEND.LT.0) GO TO 15
   10 CONTINUE
C     NUMBER OF STATIONS EXCEEDS 30
      WRITE(6,2000)
      GO TO 700
C
   15 WRITE(3,1070)
   20 CONTINUE
      READ(5,1010,END=600) NARC,IYMDB,IHB,IMB,SECB,IYMDE,IHE,IME,SECE
      WRITE (3,1011) NARC,IYMCB,IHB,IMB,SECB,IYMDE,IHE,IME,SECE
      BTIME = 3600*IHB + 60*IMB + SECB
      ETIME = 3600*IHE + 60*IME + SECE
C
C     KEY 4 IS INITIALLY NEGATIVE.IT IS SET EQUAL TO 1 AFTER THE FIRST
C     CALL TO SUBROUTINE DATAWT.
C
C
      IF (IARC.EQ.NARC) GO TO 100
C
C     A NEW ARC HAS COMMENCED.IF DATA HAS BEEN ACCUMULATED FOR THE PRE-
C     VIOUS ARC,BUT NOT YET OUTPUT (IS>0),THEN IT IS OUTPUT NOW.
C
      K = IS
      IF (K.GT.0) CALL DATAWT
      IARC = NARC
C     IF A NEW ARC HAS BEEN STARTED SET KEY4 TO 2 IN ORDER THAT
C     SUBROUTINE DATAWT SHOULD PUNCH A '2' IN KEY4 OF THE NEXT NAP '999'
C     CARD
C
      IF(KEY4.GE.0) KEY4 = 2
C
C     NOLIST IS THE NUMBER OF NONLISTED STATIONS FOR THE ARC.
C     NUMSTA IS THE NUMBER OF LISTED STATIONS (CONSTANT FOR THIS RUN).
C
      IF(NCLIST.GT.0)WRITE(6,2020)       (NONAME(I),NOSTA(I),I=1,NOLIST)
      NOLIST = 0
      DO 40 I = 1,NUMSTA
      NPASS(I) = 0
   40 CONTINUE
      WRITE(12,3000) IARC
```

```
C
C     READ DATA TAPE TILL THE TIME OF OBSERVATION IS PASSED THE START
C     TIME FOR THE ARC, THEN GO TO 200. (IYMD IS INITIALIZED TO A
C      LARGE NEGATIVE NUMBER.)
  100 CONTINUE
      IF (IYMD-IYMDB)110,120,200
  110 READ(1,1020,END=110) A,NSTA,IYMD,IH,IM,SEC,NHR,MIN,RSEC,NSIGN,
     1NDEG,NMIN,DSEC,NRED
      GO TO 100
  120 TIME = IH*3600 + IM*60 + SEC
      IF (TIME.LT.BTIME) GO TO 110
C
  200 CONTINUE
C
C     IF THE TIME OF OBSERVATION IS PASSED THE END TIME FOR THE ARC GO
C     TO 20,OTHERWISE PROCEED TO 220.
C
      IF (IYMD-IYMDE) 220,210,20
  210 IF (TIME.GT.ETIME) GO TO 20
C
  220 CONTINUE
C
C     IYMD2 IS THE CURRENT (YEAR,MONTH,DAY).IT IS INITIALIZED TO A LARGE
C     POSITIVE NUMBER.
C
      TIME = TIME + DELT
      IF (IYMD.EQ.IYMD2) GO TO 270
      IF (IS.LE.0) GO TO 260
C
C     NORDER IS AN ARRAY OF STATION NUMBERS FOR WHICH OBSERVATIONS HAVE
C     BEEN ACCUMULATED.THE STATIONS HAVE BEEN ARRANGED IN A SEQUENCE
C     SUCH THAT IF I<J THEN THE TIME OF THE FIRST ACCUMULATED (BUT NOT
C     YET OUTPUT) OBSERVATION FOR STATION NORDER(I) PRECEDES THAT FOR
C     STATION NORDER(J).
C
C     JDAY1 IS THE JULIAN DAY PRECEDING THE CURRENT ONE (JDAY2).JDAYA(L)
C     IS THE JULIAN DAY OF THE FIRST OBSERVATION OF STATION NUMBER L.
C
      DO 230 K = 1,IS
      L = NORDER(K)
      IF (JDAYA(L).GT.JDAY1) GO TO 240
  230 CONTINUE
      K = IS
      GO TO 250
```

```
  240 K = K - 1
C
C     K IS THE NUMBER OF CBSERVATIONS IN ARRAY NORDEK FOR WHICH THE DAY
C     OF THE FIRST OBSERVATION PRECEDES THE CURRENT DAY,WHICH HAS JUST
C     COME TO AN END.THE CBSERVATIONS FOR THOSE STATIONS ARE OUTPUT.
C
  250 IF (K.GT.0) CALL DATAWT
  260 CONTINUE
      JDAY1 = JDAY2
      IYR1 = IYR2
      IMO1 = IMO2
      IDA1 = IDA2
      IYMD2 = IYMD
      IYR2 = IYMD/10000
      IMO2 =  IYMD/100 - 100*IYR2
      IDA2 = IYMD - 10000*IYR2 - 100*IMO2
C     COMPUTE JULIAN DAY NUMBER
      LEAPYR = 49
      IF (IMO2.GT.2) LEAPYR = 48
      JDAY2 = (IYR2 - 50)*365 + (IYR2 - LEAPYR)/4 + MONTH(IMO2) + IDA2
C
  270 CONTINUE
      DO 280 I = 1,NUMSTA
      IF (NSTA.EQ.ISTA(I)) GO TO 290
  280 CONTINUE
      IF(NCLIST.LE.0) GO TO 284
      DO 282 I = 1,NOLIST
      IF(NSTA.EQ.NONAME(I)) GC TO 286
  282 CONTINUE
      IF(NCLIST.GE.30) GO TO 450
  284 NOLIST = NOLIST + 1
      NOSTA(NOLIST) = 1
      NONAME(NOLIST) = NSTA
      GO TO 320
  286 NOSTA(I) = NOSTA(I) + 1
      GO TO 320
C
  290 RADRA = (NHR*3600 + MIN*60 + RSEC)*SECTTR
      RADDEC = (NDEG*3600 + NMIN*60 + DSEC)*SECATR
      IF (NSIGN.EQ.MINUS) RADDEC = -RADDEC
C
  300 CONTINUE
      N = NUM(I) + 1
      IF (N.GT.1) GO TO 340
```

```
C     FIRST POINT IN STATION-CBSERVATICN ARRAY
      JDAYA(I) = JDAY2
      IS = IS + 1
      NORDER(IS) = I
  310 CONTINUE
      IF (N.LE.NMAX) GO TO 315
      WRITE(6,2030) STNAM(I),ISTA(I)
      GO TO 700
C
  315 CONTINUE
      TIM(N,I) = TIME
      RA(N,I) = RADRA
      DEC(N,I) = RADDEC
      NUM(I) = N
C
  320 READ(1,1020,END=320) A,NSTA,IYMD,IH,IM,SEC,NHR,MIN,RSEC,NSIGN,
     1NDEG,NMIN,DSEC,NRED
      TIME = IH*3600 + IM*60 + SEC
      GO TO 200
C
  340 CONTINUE
      TIMDIF = (JDAY2 - JDAYA(I))*86400 + TIME - TIM(1,I)
      IF (TIMDIF.LT.TORBT) GO TO 310
      DO 350 K = 1,IS
      IF (NORDER(K).EQ.I) GO TO 360
  350 CONTINUE
  360 CALL DATAWT
      GO TO 300
C
  450 CONTINUE
      WRITE(6,2010)         NONAME,NSTA
      GO TO 700
  600 K = IS
      IF (K.GT.0) CALL DATAWT
      KEY4 = 3
      CALL DATAWT
      IF(NOLIST.GT.0)WRITE(6,2020)        (NONAME(I),NOSTA(I),I=1,NOLIST)
      WRITE(12,1050)
  700 CONTINUE
       REWIND 1
      END FILE 2
       REWIND 2
      STOP
 1000 FORMAT (I5,5X,A8,2X,2I5)
```

```
 1001 FORMAT(11X,I5,15X,A8,12X,I5)
 1010 FORMAT(I5,5X,I6,2I3,D10.0,I6,2I3,D10.0)
 1011 FORMAT(7X,I5,4X,2(8X,2I6,I3,F9.5))
 1020 FORMAT(13A1,I5,I6,2I2,F6.4,I3,I2,F5.3,A1,2I2,F4.2,I6)
 1050 FORMAT(15X,//'JOB COMPLETED')
 1060 FORMAT(3(12X,'STATION')/10X,'NUMBER(NAP)',11X,'NAME',16X,'ID'//)
 1070 FORMAT(//10X,'ARC',19X,'START',28X,'STOP'/9X,'NUMBER',2(10X,'DATE'
     1,5X,'HR MN SECONDS'))
 1080 FORMAT (80A1)
 2000 FORMAT(15X,'NUMBER OF STATIONS EXCEEDS 30')
 2010 FORMAT('1',15X,'THE FOLLOWING STATION NUMBERS ARE NOT IN THE TABLE
     *'//3(/10(5X,I5))/5X,I5//15X,'JOB ABORTED')
 2020 FORMAT('1',15X,'THE FOLLOWING STATION NUMBERS WERE NOT IN THE TABL
     *E'//15X,'STATION NUMBER',20X,'NUMBER OF OBSERVATIONS'/30(18X,I5,27
     *X,I5/))
 2030 FORMAT(15X,A8,'-',I5,5X,'NUMBER OF DATA POINTS EXCEEDS 49')
 3000 FORMAT(////20X,'DATA SUMMARY FOR ARC NUMBER',I5//14X,'STATION',6X,
     *'DATE',7X,'GMT',9X,'PASS',5X,'NUMBER OF DATA'/12X,'ID',4X,'NAME',3
     *X,'YR MO DA',3X,'HR MN SEC',6X,'NO.',4X,'POINTS FOR ORBIT'/)
      END
      SUBROUTINE DATAWT
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/W/ TIM(49,30),RA(49,30),DEC(49,30),STNAM(30),TPASS,
     1JDAYA(30),NUM(30),NORDER(30),IYR1,IYR2,IMO1,IMO2,IDA1,IDA2,JDAY1,
     1JDAY2,IARC,IARCS,ISTA(30),JSTA(30),NPASS(30),KSTA(30),IT,K,IS
      COMMON /SAO/ M,N
      DATA I1,I201,I202,I999/1,201,202,999/
C     IT = -1 ON FIRST ENTRY, IT = 1 SUBSEQUENTLY, IT = 3 ON LAST ENTRY
C
      IF (IT.GE.3) GO TO 200
      DO 100 I = 1,K
      IF (IT.LT.0) GO TO 10
      WRITE (7,1000) I999,SNAME,IARCS,NSTAS,NPASSS,IT
   10 CONTINUE
      IT = 1
      IARCS = IARC
      M = NORDER(I)
      SNAME = STNAM(M)
      NSTAS = KSTA(M)
      NST12 = ISTA(M)
      NSTA = JSTA(M)
      WRITE (6,1010) I1,SNAME,ISTA(M),IARC
      WRITE (6,1020)
      N = NUM(M)
```

```
      NUM(M) = 0
      IF((JSTA(M)+20000).EQ.NST12) CALL SAOCOR
      DO 90 J = 1,N
      TIMP = TIM(J,M)
      IF (J.LE.1) GO TO 50
      TIMDIF = TIMP - TIM(J-1,M)
      IF (TIMDIF.GT.TPASS) GO TO 40
      IF (TIMDIF.GE.0.D0) GO TO 20
C     IF TIMDIF IS NEGATIVE THEN THE LAST OBSERVATION HAS BEEN MADE ON
C     THE DAY FOLLOWING THE PREVIOUS OBSERVATION
      IF((TIMDIF + 86400.D0).GT.TPASS) GO TO 40
C     UPDATE DATE TO NEXT DAY WHICH MUST BE DAY 2
      IYRM = IYR2
      IMOM = IMO2
      IDAM = IDA2
      DDAYM = JDAY2
   20 CONTINUE
      DELTRA = RA(J,M) - RA(J-1,M)
      DELDEC = DEC(J,M) - DEC(J-1,M)
   30 CONTINUE
      IHP = TIMP/3600.D0
      IMP = TIMP/60.D0 - 60*IHP
      SECP = TIMP - 3600*IHP - 60*IMP
      GO TO 80
   40 S = SECP + 0.01D0
      WRITE(7,1001) I202,SNAME,IARC,NSTAS,NPASS(M),IYRM,IMOM,IDAM,IHP,
     1IMP,S
      WRITE(7,1000) I999,SNAME,IARC,NSTAS,NPASS(M),IT
      WRITE (12,1040) NST12,SNAME,IYRM,IMOM,IDAM,IHP,IMP,SECP,NPASS(M)
      IF (TIMDIF.LT.0.D0) GO TO 60
      GO TO 70
   50 CONTINUE
      IF (JDAYA(M).GT.JDAY1) GO TO 60
      IYRM = IYR1
      IMOM = IMO1
      IDAM = IDA1
      DDAYM = JDAY1
      GO TO 70
   60 IYRM = IYR2
      IMOM = IMO2
      IDAM = IDA2
      DDAYM = JDAY2
   70 CONTINUE
      DELTRA = 0.D0
```

```
      DELDEC = 0.D0
      IHP = TIMP/3600.D0
      IMP = TIMP/60.D0 - 60*IHP
      SECP = TIMP - 3600*IHP - 60*IMP
      S = SECP - 0.01D0
      NPASS(M) = NPASS(M) + 1
      WRITE (7,1001) I201,SNAME,IARC,NSTAS,NPASS(M),IYRM,IMOM,IDAM,IHP,
     1IMP,S
   80 CONTINUE
      WRITE (6,1030) IYRM,IMOM,IDAM,IHP,IMP,SECP   ,NPASS(M),RA(J,M),
     1DELTRA,DEC(J,M),DELDEC
      WRITE(2) NSTA,DDAYM,TIMP,RA(J,M),DEC(J,M)
   90 CONTINUE
      S = SECP + 0.01D0
      WRITE(7,1001)I202,SNAME,IARC,NSTAS,NPASS(M),IYRM,IMOM,IDAM,IHP,
     1IMP,S
      NPASSS = NPASS(M)
      WRITE (12,1040) NST12,SNAME,IYRM,IMOM,IDAM,IHP,IMP,SECP,NPASS(M),N
  100 CONTINUE
      IS = IS - K
      IF(IS.LE.0) GO TO 130
      DO 120  I = 1,IS
      NORDER(I) = NORDER(I + K)
  120 CONTINUE
  130 CONTINUE
      RETURN
  200 WRITE(7,1000)I999,SNAME,IARCS,NSTAS,NPASSS,IT
      RETURN
 1000 FORMAT (I3,2X,A8,4I3)
 1001 FORMAT (I3,2X,A8,3I3,6X,5I3,F15.3)
 1010 FORMAT(I1,9X,A8,' - ',I5,15X,'ARC NUMBER',I8//)
 1020 FORMAT(4X,'DATE',7X,'GMT',9X,'PASS',3X,'RIGHT ASCENSION',4X,'DELTA
     1 RA',9X,'DECLINATION',8X,'DELTA DEC'/2X,'YR MO DA',2X,'HR MN SEC '
     1,5X,'NO.',5X,'(RADIANS)',8X,'(RADIANS)',9X,'(RADIANS)',9X,'(RADIAN
     1S)'/)
 1030 FORMAT(X,3I3,I4,I3,F8.4,I4,4(3X,F15.10))
 1040 FORMAT(10X,I5,X,A8,3I3,I5,I3,F4.0,I8,8X,I5)
      END
      SUBROUTINE TIMCOR
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/T/  TIM,COR,STHETA,CTHETA,ZETA,Z
      COMMON/M/ TIM2
      DIMENSION MONTH(12)
      DATA MONTH/-1,30,58,89,119,150,180,211,242,272,303,333/
```

```
      DATA TIM2IN /-1.D+30/
    5 TIMDIF = TIM - TIM2
      IF (TIMDIF.LE.0.D0) GO TO 15
   10 COR1 = COR2
      TIM1 = TIM2
      IYR1 = IYR2
      IMO1 = IMO2
      IDA1 = IDA2
      READ (4,30,END=20) IYR2,IMO2,IDA2,COR2
      WRITS (3,60) IYR2,IMO2,IDA2,COR2
      LEAPYR = 49
      IF (IMO2.GT.2) LEAPYR = 48
      TIM2 = (IYR2 - 50)*365 + (IYR2 - LEAPYR)/4 + MONTH(IMO2) + IDA2
      IF (TIM1.LE.TIM2IN) GO TO 10
      F = (COR2 - COR1)/(TIM2 - TIM1)
      GO TO 5
   15 COR = COR2 + F*TIMDIF
      RETURN
   20 WRITE (6,50)
      STOP
   30 FORMAT (3I2,4X,F10.5)
   50 FORMAT (X,'END OF DATA SET 4,(TIME CORRECTIONS).JOB ABORTED.')
   60 FORMAT (90X,3I3,2X,'A.1 - UTC = ',F10.5)
      END
      SUBROUTINE PRENUT
C
      IMPLICIT REAL*8(C-H,O-Z)
C
      COMMON/T/ DAYS,COR,SHP,CHP,UP,ZP
      DIMENSION A(23),AS(30),ADP(30),ADE(30),ADELE(23),ADELP(23)
C
C
      DATA ADELE/0.,0.,-2.,2.,2.,-2.,3.,3.,0.,5.,7.,8.,0.,0.,0.,-24.,-66
     *.,0.,0.,0.,0.,0.,0./
C
      DATA ADELP/-2.,-3.,3.,-3.,-4.,4.,-5.,-5.,10.,-10.,-15.,-15.,16.,-2
     *1.,45.,45.,0.,0.,0.,0.,C.,0.,0./
C
C
      DATA ADE/-3.,3.,3.,-2.,3.,3.,3.,0.,-3.,0.,5.,7.,-7.,-10.,0.,11.,-1
     *1.,0.,14.,23.,22.,30.,-31.,0.,-50.,0.,0.,183.,0.,0./
C
      DATA ADP/5.,-5.,-5.,6.,-6.,-6.,-6.,6.,7.,-7.,-9.,-13.,14.,19.,25.,
     *-26.,26.,28.,-32.,-44.,-52.,-57.,58.,60.,114.,-149.,-261.,0.,0.,0.
```

```
      */
C
      DATA STRAD,DPLUS/.4848136811D-5,18262.5D0/
      DATA ADR/57.2957795/
C
C
      DD = DPLUS + DAYS
      TD = DD/36525.D0
      AT = TD
      ADD = DD/10000.D0
      ADD2 = ADD * ADD
      ELD = 13.0649924465D0*DD + 296.104608D0
      CALL PRINCP(ELD,AL)
      ELPD = .9856002669D0*DD + 358.475833D0
      CALL PRINCP(ELPD,ALP)
      EFD = 13.2293504490D0*DD + 11.250889D0
      CALL PRINCP(EFD,AF)
      DED = 12.1907491914D0*DD + 350.737486D0
      CALL PRINCP(DED,AD)
      OMEGD = .0529539222D0*DD - 259.183275D0
      CALL PRINCP(OMEGD,AMEG)
      AL = (AL +(.0006890    + .000000295*ADD) * ADD2)/ADR
      ALP = (ALP -(.0000112 + .000000068 * ADD) * ADD2)/ADR
      AF = (AF - (.0002407  + .000000007 * ADD) * ADD2)/ADR
      AD = (AD - (.0001076 - .000000039 * ADD) * ADD2)/ADR
      AMEG = ((.0001557  + .000000046 * ADD) * ADD2 - AMEG)/ADR
      A(23) = AMEG
      A(21) = AMEG + AMEG
      A(9) = 2. * (AL - AF)
      A(16) = AMEG - A(9)
      A(5) = 2. * (AF - ALP- AD) + AMEG
      A(4) = A(16) + AMEG
      A(1) = AL - ALP - AD
      A(22) = 2. * (AF-AD+AMEG)
      A(20) = ALP
      A(19) = ALP + A(22)
      A(18) = A(22) - ALP
      A(17) = A(22) - AMEG
      A(15) = 2. * (AL-AD)
      A(14) = 2. * (AF-AD)
      A(13) = ALP + ALP
      A(12) = ALP + AMEG
      A(11) = A(12) + A(12) + A(14)
      A(10) = AMEG - ALP
```

```
      A(8) = AMEG - A(15)
      A(7) = A(10) + A(14)
      A(6) = A(15) + AMEG
      A(3) = A(12) + A(14)
      A(2) = AL - AD
      AS(30) = 2. * (AF + AMEG)
      AS(29) = AL
      AS(28) = AS(30) - AMEG
      AS(27) = AS(30) + AL
      AS(26) = AL - AD - AD
      AS(25) = AS(30) - AL
      AS(24) = AD + AD
      AS(23) = AL + AMEG
      AS(22) = AMEG - AL
      AS(21) = AS(24) + AS(25)
      AS(20) = AS(27) - AMEG
      AS(19) = AS(30) + AS(24)
      AS(18) = AL + AL
      AS(17) = AS(27) - AS(24)
      AS(16) = AS(30) + AS(18)
      AS(15) = AF + AF
      AS(14) = AS(22) + AS(15)
      AS(13) = AS(22) + AS(24)
      AS(12) = AS(23) - AS(24)
      AS(11) = AS(14) + AS(24)
      AS(10) = AS(26) + ALP
      AS(9) =  AS(30) + ALP
      AS(8)  = AL + AS(24)
      AS(7) = AMEG + AS(24)
      AS(6) = AS(30) - ALP
      AS(5) = AS(30) + AS(8)
      AS(4) = AS(17) + AL
      AS(3) = AMEG - AS(24)
      AS(2) = AS(19) - AMEG
      AS(1) = AS(17) - AMEG
      ASS16   = AD
      ASS15   = AS(10) - AL
      ASS14   = AL - ALP
      ASS13   = AL - AF -AF
      ASS12 = AS(16) - AMEG
      ASS11   = AL + AF + AF
      ASS10   = AL + ALP
      ASS9   = AS(30) + ASS14
      ASS8   = AMEG - AL -AL
```

```
      ASS7   = AS(14) - AD -AD
      ASS6   = AL + AL + AMEG
      ASS5   = AS(25) - ASS15
      ASS4   = ASS5   + AL
      ASS3   = AL + AMEG + AMEG
      ASS2   = AS(27) + ALP
      ASS1   = AS(27)+ AL +AL
      ADELP(23) = -172327. - 173.7*AT
      ADELP(21) = 2088. + 0.2 * AT
      ADELP(22) = - 12729. - 1.3 * AT
      ADELP(20) = 1261. - 3.1 * AT
      ADELP(19) = -497. + 1.2 * AT
      ADELP(18) = 214. - C.5 * AT
      ADELP(17) = 124. + C.1 * AT
      ADELP(13) = 16. - 0.1 * AT
      ADELP(11) = -15. + C.1 * AT
      ADELE(23) = 92100. + 9.1 * AT
      ADELE(21) =  - 904. + 0.4 * AT
      ADELE(22) = 5522. - 2.9 * AT
      ADELE(19) = 216. - C.6 * AT
      ADELE(18) = -93. + 0.3 * AT
      ADP(30) = -2037. - C.2 * AT
      ADP(29) = 675. + 0.1 * AT
      ADP(28) = -342. - 0.4 *AT
      ADE(30) = 884. - 0.5 * AT
      ADE(27) = 113. - 0.1 * AT
      BDELE = 0.
      BDELP = 0.
      DO 100 I = 1,23
      BDELE = BDELE + ADELE(I) * COS(A(I))
      BDELP = BDELP + ADELP(I) * SIN(A(I))
  100 CONTINUE
      BDE = 2. * COS(ASS12)
      BDP = 2.*(-SIN(ASS1)+SIN(ASS2)-SIN(ASS3)-SIN(ASS4)-SIN(ASS5)
     1         +SIN(ASS6)-SIN(ASS7)-SIN(ASS8))     +
     2      3.*(-SIN(ASS9)-SIN(ASS10)+SIN(ASS11))     +
     3      4.*(-SIN(ASS12)+SIN(ASS13)+SIN(ASS14)-SIN(ASS15)-SIN(ASS16))
      DO 200 I = 1,30
      BDE = BDE + ADE(I) * COS(AS(I))
      BDP = BDP + ADP(I) * SIN(AS(I))
  200 CONTINUE
      DELTAE = (BDE + BDELE)/36000000.
      BDTAP = (BDP + BDELP)/1C000.
      EPSD=23.452294DO - TD*(0.0130125DO + TD*(0.0000164DO -
```

```
     1TD*0.00000050300)) + DELTAE
      T50 = DAYS/36525.D0
      TDRAD = T50 * STRAD
C
C      U = ZETA,   H = THETA ,   UAZ = ZETA + Z
C
      U     = TDRAD*(2304.997D0 +T50*(.302D0 +T50*.0179D0))
      Z = TDRAD*(2304.997D0 +T50*(1.093D0 +T50*.0192D0))
      H     = TDRAD*(2004.298D0 -T50*(.0426D0 +T50*.0416D0))
      XPSI = BDTAP*STRAD
      XEPS = DELTAE/ADR
      EPSO = EPSD/ADR
      CZ = DCOS(Z)
      SZ = DSIN(Z)
      SH = DSIN(H)
      CH = DCOS(H)
      CEPSD = DCOS(EPSD)
      SEPSD = DSIN(EPSD)
      EPSO = EPSO - XEPS
      CEPSO = DCOS(EPSO)
      SEPSO = DSIN(EPSO)
      SXPSI = DSIN(XPSI)
      CXPSI = DCOS(XPSI)
      SU1 = SEPSO*CEPSD - CEPSO*SEPSD*CXPSI
      CU1 = SEPSD*SXPSI
      SZ1 = SEPSD*CEPSO - CEPSD*SEPSO*CXPSI
      CZ1 = SEPSO*SXPSI
      CH1 = CEPSD*CEPSO + SEPSD*SEPSO*CXPSI
       Z1 = DATAN2(SZ1,CZ1)
       CUAZ = CZ * CU1 - SZ*SU1
       CHP = CH*CH1 - SH*CUAZ
      CUPMU = SH*CH1 + CH*CUAZ
       SUPMU = CZ*SU1 + SZ*CU1
       UPMU = DATAN2(SUPMU,CUPMU)
      UP = U + UPMU
      CUPLU = DCOS(UPMU)
      SUPLU = DSIN(UPMU)
      SHP = CUPMU*CUPLU + SUPMU*SUPLU
      CZPMZ = CH*SHP - SH*CHP*CUPLU
      SZPMZ = SH * SUPLU
      ZP = Z1+ DATAN2(SZPMZ,CZPMZ)
      RETURN
      END
      SUBROUTINE PRINCP(XDOUBL,X)
```

```
      REAL*8 XDOUBL,XT
      I = XDOUBL
      XT = ((I + 180)/360)*36C
      XDOUBL = XDOUBL - XT
      X = XDOUBL
      RETURN
      END
      SUBROUTINE SAOCOR
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMCN/W/ TIM(49,30),RA(49,30),DEC(49,30),STNAM(30),TPASS,
     1JDAYA(30),NUM(30),NCRDER(30),IYR1,IYR2,IMO1,IMO2,IDA1,IDA2,JDAY1,
     1JDAY2,IARC,IARCS,ISTA(3C),JSTA(30),NPASS(30),KSTA(30),KEY4,K,IS
      COMMCN/SAO/ M,N
      COMMCN/T/ DAYS,COR,STHETA,CTHETA,ZETA,Z
      DATA PI,TWOPI/3.14DC,6.283185307D0/
      DAYS = JDAYA(M) + TIM(1,M)/86400.D0
      CALL PRENUT
      CALL TIMCOR
      DO 5C0 I = 1,N
      UARO = ZETA + RA(I,M)
      CDECO = DCOS(DEC(I,M))
      SDECO = DSIN(DEC(I,M))
      SRMZ = CDECO * DSIN(UARC)
      DUM = CDECO * DCOS(UARO)
      CRMZ = CTHETA*DUM - STHETA*SDECO
      SDEC = CTHETA*SDECO + STHETA*DUM
      RMZ = DATAN2(SRMZ,CRMZ)
      CDEC = SRMZ * DSIN(RMZ) + CRMZ*DCOS(RMZ)
      DECF = DATAN2(SDEC,CDEC)
      AF = RMZ + Z
      IF (DECF.GT.PI) DECF = DECF - TWOPI
      IF (AF.GE.TWOPI) AF = AF - TWOPI
      RA(I,M) = AF
      DEC(I,M) = DECF
      TIM(I,M) = TIM(I,M) - CCR
C     THIS IS A TEMPORARY CHANGE TO FIX ERROR IN SAO TIME
      ITIME=TIM(I,M)
      TIM(I,M)=ITIME
  500 CONTINUE
      RETURN
      END
```

```
//Z7GEMOPB JOB (G70041150A,T,D00138,H00H00),69,MSGLEVEL=1
//   EXEC  LOADER,PARM='MAP,CALL,SIZE=150K',REGION.GO=160K
//GO.FT06F001 DD  SYSOUT=A,SPACE=(CYL,(8,2))
//GO.FT07F001 DD  SYSOUT=B,DSN=DECK
//GO.SYSLIN    DD UNIT=2400-9,LABEL=(5,BLP),DISP=(NEW,KEEP),
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),VOL=SER=33599C
//GO.FT01F001 DD  UNIT=2400-9,LABEL=(3,BLP),DISP=(OLD,DELETE),
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200,DEN=2),VOL=SER=3826F
//GO.FT02F001 DD  DUMMY
//GO.FT03F001 DD  SYSOUT=A,SPACE=(CYL,(2,2))
//GO.FT12F001 DD  SYSOUT=A,SPACE=(CYL,(2,2))
//GO.SYSABEND DD  SYSOUT=A,SPACE=(CYL,(1))
//GO.DATA5 DD *
      9      ORGAN      29001
     19      SPAIN      29004
     10      JUPTR      29010
     50      COLDLK     29424
     11      EDWAFB     29425
      3      BPOIN      31021
      1      FTMYR      31022
     47      OOMER      31024
     48      SATAG      31028
      2      MOJAV      31030
     68      NEWFL      31032
     52      COLEG      31033
      4      GFORK      31034
     25      ROSMA      31042
      6      EDINB      37036
      5      COLBA      37037
     14      BERMD      37039
     16      PURIO      37040
      7      GSFCP      37043
     26      DENVR      37045
     32      JUP24      37071
     40      JUP40      37072
     53      JUPTH      37073
     33      JBC4L      37074    -1
      2      660108   0  0            660115  0  0
/*
//GO.FT04F001 DD *
651225       4.3395
661225       4.3576
/*
```

0043 CARDS

A-2.6 PRENAP CARD UPDATER PROGRAM AND SAMPLE JCL

```
C     UPDATER ROUTINE FOR NAP-2 INPUT CARDS.
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL*1  A,B,BLANK
      DIMENSION A(80),B(80),M(80),KAT(2),KEY(10),DATA(2)
      EQUIVALENCE (B(1),BETA)
      DATA  INS,REP,DEL,MCD/' INS ','REP ','DEL ','MOD '/
      DATA BLANKS,BLANK/'            ',' '/
      INTEGER*4 ALPHA,INS,REP,DEL
 1000 FORMAT (A4,X,I5,5X,I5)
 1001 FORMAT (10X,A4,X,I5,5X,I5)
 1010 FORMAT (10X,'UPDATE ERRCR  ',A4)
 2020 FORMAT (I3,I2,A8,10I3,D22.15,D15.8)
 2010 FORMAT (80A1)
 2011 FORMAT (10X,80A1)
      WRITE (6,3000)
      WRITE (6,3010)
 3000 FORMAT ('1',30X,'PRCGRAM TO UPDATE NAP INPUT CARDS STORED ON TAPE'
     *///3X,'INS  I1',9X,'INSERTS CARDS AFTER SEQUENCE NUMBER I1.'/3X,'D
     *EL  I1  I2      DELETES CARDS BETWEEN (AND INCLUDING) SEQUENCE NUMB
     *ERS I1 AND I2.'/3X,'REP  I1  I2      DELETES CARDS BETWEEN (AND INC
     *LUDING) SEQUENCE NUMBERS I1 AND I2,AND THEN INSERTS CARDS.'/3X,'MO
     *D  I1  I2      MODIFIES CARDS BETWEEN (AND INCLUDING) SEQUENCE NUMB
     *ERS I1 AND I2.'//)
 3010 FORMAT (22X,'FORMAT FOR ABOVE FOUR CARDS (A4,X,I5,5X,I5)  NOTE THA
     *T INS ,DEL ,REP , AND MCD  START IN'/17X,'COLUMN 1 AND THAT COLUMN
     * 4 SHOULD BE BLANK.'//22X,'INS AND REP MUST BE FOLLOWED BY THE CAR
     *DS TO BE INSERTED.  THE CARDS MUST'/17X,'BE FCLLOWED BY A BLANK CA
     *RD TO SIGNIFY THE END OF THE PARTICULAR SET OF INSERTED CARDS.'//2
     *2X,'THE MOD CARDS MUST BE FOLLOWED BY A MODIFYING CARD.  COLUMNS N
     *OT TO BE MODIFIED'/17X,'SHOULD BE LEFT BLANK.'//1X,'**************
     ****************************************************************
     ******************************************'//)
      I=0
   10 READ (5,1000,END = 100) ALPHA, I1,I2
      WRITE (6,1001) ALPHA,I1,I2
      IF (ALPHA.EQ.INS) GC TO 20
      IF (ALPHA.EQ.REP) GC TO 30
      IF (ALPHA.EC.DEL) GC TO 30
      IF(ALPHA.EQ.MOD) GO TO 30
      WRITE (6,1010) ALPHA
      STOP
   20 IF (I.GE.I1) GO TO 40
      READ(9,END=200) KAT,XLABL,KEY,DATA,I
      WRITE (12,2020) KAT, XLABL,KEY,DATA
```

```
      GO TC 20
   30 I1=I1-1
      GO TO 20
   40 IF (ALPHA.EQ.DEL) GC TO 70
      IF (ALPHA.EQ.MOD) GC TO 3CO
   50 READ (5,2010) B
      WRITE (6,2011) B
      IF (BETA.EQ.BLANKS) GO TO 60
      WRITE (12,2010) B
      GO TO 50
   60 IF (ALPHA.EQ.INS) GC TO 10
   70 IF (I.GE.I2) GO TO 10
      READ(9,END=200) KAT,XLABL,KEY,DATA,I
      GO TC 70
  100 READ (9,END=200) KAT,XLABL,KEY,DATA,I
      WRITE(12,2020) KAT,XLABL,KEY,DATA
      GC TO 100
  200 STOP
  300 READ (5,2010) B
      WRITE(6,2011) B
      L=0
      DO 31C J=1,80
      IF(B(J).EQ.BLANK) GO TO 310
      L=L+1
      M(L)=J
  310 CONTINUE
  320 CONTINUE
      IF(I.GE.I2) GO TO 1C
      REWIND 10
      READ (9,END=200) KAT,XLABL,KEY,DATA,I
      WRITE (10,2020) KAT,XLABL,KEY,DATA
      REWIND 10
      READ (10,2010) A
      DO 330  K=1,L
      J=M(K)
  330 A(J)=B(J)
      WRITE(12,2010) A
      GO TO 320
      END
```

0083 CARDS

```
// EXEC LOADER,PARM='MAP,CALL,SIZE=100K',REGION.GO=110K
//GO.SYSLIN DD *

********** PRENAP CARD UPDATER OBJECT DECK IS INSERTED HERE     ********

/*
//GO.FT09F001 DD UNIT=9TRACK,LABEL=(3,BLP),DSN=ITAPEGEM,
// DCB=(RECFM=VBS,LRECL=80,BLKSIZE=3204),DISP=(OLD,PASS),
// VOL=SER=5630
//GO.FT10F001 DD UNIT=DISK,SPACE=(TRK,(1,1)),
// DCB=(RECFM=VBS,LRECL=84,BLKSIZE=172),DISP=(NEW,DELETE)
//GO.FT12F001 DD UNIT=DISK,SPACE=(CYL,(3,2)),DSN=NAPOLEON,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),DISP=(NEW,PASS)
//GO.DATA5 DD *
INS     12
999 OMOJAV6E   2 25 22  1  0  0  0  0  0  0         0.0            0.0

DEL     30          35
REP     36          40
999 OGFORK6E   2 27 21  1  0  0  0  0  0  0         0.0            0.0
201 OFTMYR6E   2  1 22  0  0 66  1 34  0  0         0.51500000D 03 0.0
202 OFTMYR6E   2  1 22  0  0 66  1 34  0  0         0.57780000D 03 0.0
701 1FTMEQL    1  1  6  3          1                0.10000000D 01 0.10000000D 16
701 1FTMEQM    2  1  7  4          1                0.10000000D 01 0.10000000D 16

MOC     50          55
                         0
/*
                                                                      0034 CARDS
```

A-2.7 POST NAP PROGRAM

```
C     NAP-2 POST PROCESSING PROGRAM
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON  /U601/ PLOB,EMPLOB,ABIAS(60),XLABL,DATA(2),KAT,KONT,
     *KEY(10)
      DIMENSION KEY704(9),KL701(8),KM701(8),X704(2),X701(2),K704(2),
     *K701(2),KSEVEN(2)
      DATA X1,D15/1.D0,1.D15/
      DATA K701/701,1/
      DATA K704/704,1/
      DATA BLANK/0.D0/
      DATA KEY704,KL701,KM701/9*0,6,3,6*0,7,4,6*0/
      WRITE (6,1107)
 1107 FORMAT (6X,'DATA SET 5 INPUT IOPT.,MAXERR, SDMAX, FREQCY ACCORDING
     * TO FORMAT (2I5,2D10.1)  FREQCY=136.83(GEOS I)'/1X,'FREQCY=136.32(
     *GEOS II)'//6X,'SDMAX. IF THE MEASUREMENT STANDARD DEVIATION (IN RA
     *DIANS) FOR A PASS EXCEEDS SDMAX THEN THE MEASUREMENTS FOR'/1X,'THA
     *T PASS ARE EFFECTIVELY WEIGHTED OUT OF THE SOLUTION.'//6X,'MAXERR.
     *  IF THE MEAN MEASUREMENT ERROR (IN MINITRACK COUNT) FOR A PASS EX
     *CEEDS MAXERR THEN THE MEASUREMENTS FOR'/1X,'THAT PASS ARE EFFECTIV
     *ELY WEIGHTED OUT OF THE SOLUTION.'//6X,'IF THE MEASUREMENTS FOR A
     *PASS HAVE NOT BEEN WEIGHTED OUT OF THE SOLUTION---SEE SDMAX AND MA
     *XERR ABOVE---AND'/)
      WRITE (6,1108)
 1108 FORMAT (//1X,'IOPT=1',10X,'THE APRIORI MEASUREMENT SIGMA IN THE PA
     *SS IS GIVEN AS 10 TIMES THE OBSERVED STANDARD DEVIATION.'/17X,'IF
     *THERE IS NO OBSERVED STANDARD DEVIATION (ONE MEASUREMENT ONLY) BUT
     * THERE EXISTS A PREVIOUS 701'/17X,'CARD THEN THAT CARD IS COPIED.
     * IF THERE IS NO PREVIOUS 701 CARD, THEN THE MEASUREMENT IS WEIGHTE
     *D'/17X,'OUT OF THE SOLUTION.'//1X,'IOPT=0',10X,'THE APRIORI MEASUR
     *EMENT SIGMA IS CONSTANT FOR ALL PASSES AND NO 701 CARD IS OUTPUT F
     *OR THE PASS,'/17X,'UNLESS THERE IS NO OBSERVED STANDARD DEVIATION.
     *  IF THERE IS NO OBSERVED STANDARD DEVIATION, BUT'/17X,'THERE EXIS
     *TS A PREVIOUS 701 CARD, THEN THAT CARD IS COPIED.  IF NO PREVIOUS
     *701 CARD EXISTS THEN NO'/17X,'701 CARD IS OUTPUT.'/1X,'IOPT=-1',9X
     *,'THE APRIORI MEASUREMENT SIGMA IS CONSTANT FOR ALL PASSES AND NO
     *701 CARD IS OUTPUT FOR THE PASS,'/17X,'UNLESS THERE EXISTS NO OBSE
     *RVED STANDARD DEVIATION.  IF THERE IS NO OBSERVED STANDARD DEVIATI
     *ON,  BUT'/17X,'THERE EXISTS A 701 CARD, THEN THAT CARD IS COPIED.
     * IF NO PREVIOUS 701 CARD EXISTS THEN THE'/17X,'MEASUREMENT IS WEIG
     *HTED OUT OF THE SOLUTION.'//)
      WRITE (6,1109)
 1109 FORMAT (1X,'00000000000000000000000000000000000000000000000000000000000000000000000
     *0000000000000000000000000000000000000000000000000000000000000000000000000000000'///)
 1110 FORMAT(I3,I2,A8,10I3,D22.8,D15.8)
```

```
      KOUNT = 0
C     READ OPTION CARD FOR 701 CONTINUATION 1 CARDS
C     IF THE STANDARD DEVIATION OF A SET OF MEASUREMENTS FOR A PASS
C     (SDEVL AND SDCUM) EXCEEDS SDMAX OR IF THE MEAN MEASUREMENT ERROR
C     IN MINTRACK COUNTS EXCEEDS MAXERR THEN THE MEASUREMENT IS
C     EFFECTIVELY WEIGHTED OUT OF THE SOLUTION.
C     IOPT = -1,0,OR 1  IF IOPT = 1 THEN (UNLESS ACCORDING TO THE ABOVE
C     CRITERION THE MEASUREMENT HAS BEEN WEIGHTED OUT OF THE SOLUTION)
C     THE A PRIORI MEASUREMENT SIGMA FOR THE PASS IS GIVEN AS
C     10 TIMES THE OBSERVED STANDARD DEVIATION FOR THE PASS.
C             NOTE THAT  SDEVL = 0.DO DOES NOT INDICATE THAT
C     THE STANDARD DEVIATION EQUALS ZERO, BUT RATHER THAT NO STANDARD
C     DEVIATION HAS BEEN COMPUTED FOR THIS NAP RUN. IF THAT IS THE
C     CASE, AUD A 701, CONTINUATION 1, FOR THS L-MEASUREMENT WAS USED
C     IN THIS RUN THEN THAT CARD IS COPIED (UNLESS THE MEASUREMENT
C     HAS BEEN WEIGHTED OUT BECAUSE OF AN EXCESSIVE ERROR -SEE ABOVE)
C     HOWEVER,IF NO PREVIOUS CARD EXISTS AND IOPT IS NOT EQUAL TO ZERO
C     THEN THE MEASUREMENT IS EFFECTIVELY WEIGHTED OUT. THE SAME APPLIES
C     TO THE M-MEASUREMENTS.
C     FREQCY = 136.83DO   (GEOS I)
C            = 136.32DO  (GEOS II)
C
      READ (5,1100) IOPT, MAXERR, SDMAX, FREQCY
      WRITE(6,1101) IOPT,MAXERR,SDMAX,FREQCY
C
      WRITE (31,1102)
      I=0
    2 READ(10) N,X,Y
      IF(N.NE.601) GO TO 4
      I=I+1
      WRITE(12) N,X,Y
      GO TO 2
    4 REWIND 10
      REWIND 12
      DO 6 J=1,I
    6 READ(10)
C
      PLOB=136.DO/(FREQCY*57.DO)
      PCNT=PLOB/1000.DO
      EMPLOB=136.DO/(FREQCY*46.DO)-PLOB
      EMPCNT=EMPLOB/1000.DO
C
C     READ NAP CONTROL CARDS
C
```

```
   10 READ(27,1110,END=999) KAT,KONT,XLABL,KEY,DATA
      IF(KAT.EQ.601) CALL UP601
   20 KOUNT=KOUNT+1
      WRITE(8) KAT,KONT,XLABL,KEY, DATA,KOUNT
      WRITE(6,1000) KOUNT,KAT,KONT,XLABL,KEY,DATA
      IF(KAT.NE.202) GO TO 10
C
C     PROCESS PASS
C
  100 X701(1)=0.D0
      X701(2)=0.D0
      X704(1)=0.D0
      X704(2)=0.D0
      KSEVEN(1)=0
      KSEVEN(2)=0
      SL701=-1.D0
      SM701=-1.D0
      SL704=0.D0
      SM704=0.D0
C
C     XK=0 OR 1 AND CNT= PCNT OR ECNT DEPENDING ON WHETHER THE STATION
C     IS POLAR(EVEN) OR EQUATORIAL(ODD)
C
      SEC=DATA(1)
      IDAY=KEY(8)
      NSTATN=KEY(2)
      LF= NSTATN+30
      XK=NSTATN-NSTATN/2*2
      CNT=PCNT+EMPCNT*XK
C
      READ(10) XLABLM,MN2,NUMPT1,NUMPT2,XLABLL,XMEANL,SDEVL,XMEANM,SDEVM
      MN1=MN2-1
  110 READ(27,1110,END=999) KAT,KONT,XLABL,KEY,DATA
      IF(KAT.EQ.999) GO TO 200
C
C     LM12 = 1 OR 2 ACCORDING AS KEY(1) IS ODD(L) OR EVEN(M)
C
      LM12=2-KEY(1) + KEY(1)/2*2
      IF(KAT.EQ.701) GO TO 130
      IF(KAT.EQ.704) GO TO 140
  120 KOUNT = KOUNT + 1
      WRITE(8) KAT,KONT,XLABL,KEY,DATA,KOUNT
      WRITE(6,1000) KOUNT,KAT,KONT,XLABL,KEY,DATA
      GO TO 110
```

```
C
C     701 CARD
  130 X701(LM12) = DATA(2)
      KSEVEN(LM12)=KEY(7)
      GO TO 110
C
C     704 CARD
  140 X704(LM12) = DATA(1)
      GO TO 110
C     999 CARD
C
C
  200 ERRORL=(XMEANL-X704(1)+ABIAS(MN1))/CNT
      IF(ERRORL.LT.0.D0) GO TO 210
      LERROR=ERRORL + 0.5D0
      LLOBE=-(LERROR + 500)/1000
      GO TO 220
  210 LERROR=ERRORL-0.5D0
      LLOBE= -(LERROR-500)/1000
  220 ERRORM=(XMEANM-X704(2)+ABIAS(MN2))/CNT
      IF(ERRORM.LT.0.D0) GO TO 230
      MERROR=ERRORM + 0.5D0
      MLOBE=-(MERROR + 500)/1000
      GO TO 240
  230 MERROR = ERRORM -0.5D0
      MLOBE=-(MERROR-500)/1000
  240 IF(LLOBE.EQ.0) GO TO 250
      LERROR=LERROR + 1000*LLOBE
      SL704=1000*LLOBE
      SL704=SL704*CNT
      KOUNT=KOUNT + 1
      WRITE(8) K704,XLABLL,MN1,KEY704,SL704,BLANK,KOUNT
      WRITE(6,1000) KOUNT,K704,XLABLL,MN1,KEY704,SL704
  250 IF(MLOBE.EQ.0) GO TO 255
      MERROR=MERROR + 1000*MLOBE
      SM704=1000*MLOBE
      SM704=SM704*CNT
      KOUNT=KOUNT + 1
      WRITE (8) K704,XLABLM,MN2,KEY704,SM704,BLANK,KOUNT
      WRITE(6,1000) KOUNT,K704,XLABLM,MN2,KEY704,SM704
C
C
C
  255 IF(SDEVL.LE.0.D0) GO TO 320
```

```
  260 IF(IABS(LERROR).GT.MAXERR)    GO TO 330
      IF(SDEVL.LE.0.DO) GO TO 300
      IF(SDEVL.GT.SDMAX)    GO TO 340
      IF(IOPT.LE.0) GO TO 355
      SL701= SDEVL*10.DO
      KL701(5)=0
  280 KOUNT=KOUNT + 1
      WRITE(8) K701,XLABLL,MN1,NSTATN,KL701,X1,SL701,KOUNT
      WRITE(6,1000) KOUNT,K701,XLABLL,MN1,NSTATN,KL701,X1,SL701
      GO TO 355
C
  290 SL701=D15
  295 KL701(5)=1
      GO TO 280
  300 IF(X701(1).GT.0.DO) GO TO 290
      IF(IOPT.NE.0) GO TO 290
      GO TO 355
  320 IF(X701(1).GE.D15) GO TO 260
      SDEVL=X701(1)*1.D-1
      IF(KSEVEN(1).GT.0) SDEVL=SDEVL*1.D-6
      GO TO 260
  330 IF(SDEVL.LE.0.DO) GO TO 290
  340 IF(IOPT.LE.0) GO TO 290
      SL701=SDEVL*1.D7
      GO TO 295
C
  355 IF(SDEVM.LE.0.DO) GO TO 420
  360 IF(IABS(MERROR).GT.MAXERR) GO TO  430
      IF(SDEVM.LE.0.DO) GO TO 400
      IF(SDEVM.GT.SDMAX) GO TO 440
      IF(IOPT.LE.0) GO TO 500
      SM701=SDEVM * 10.DO
      KM701(5)= 0
  380 KOUNT=KOUNT + 1
      WRITE(8)K701,XLABLM,MN2,NSTATN,KM701,X1,SM701,KOUNT
      WRITE (6,1000) KOUNT,K701,XLABLM,MN2,NSTATN,KM701,X1,SM701
      GO TO 500
  390 SM701=D15
  395 KM701(5)=1
      GO TO 380
  400 IF(X701(2).GT.0.DO) GO TO 390
      IF(IOPT.NE.0) GO TO 390
      GO TO 500
  420 IF(X701(2).GE.D15) GO TO 360
```

```
      SDEVM=X701(2)*1.D-1
      IF(KSEVEN(2).GT.0) SDEVM=SDEVM*1.D-6
      GO TO 360
  430 IF(SDEVM.LE.0.D0) GO TO 390
  440 IF(IOPT.LE.0) GO TO 390
      SM701=SDEVM*1.D7
      GO TO 395
C
  500 LSD=SDEVL/CNT+0.5D0
      MSD=SDEVM/CNT+0.5D0
      WRITE(11) LF,IDAY,SEC,KEY(1),KEY(3),LERROR,MERROR,LLOBE,MLOBE,
     * LSD,MSD,KL701(5),KM701(5),KSEVEN
      GO TO 20
C
C
C
  999 REWIND 11
      READ(10,END=998)
  998 CALL IWRITE
      REWIND 8
  997 READ(8,END=996) KAT,KONT,XLABL,KEY,DATA,KOUNT
      WRITE(10)       KAT,KONT,XLABL,KEY,DATA,KOUNT
      GO TO 997
  996 CONTINUE
      WRITE(6,1006)
      STOP
 1000 FORMAT(2I8,I2,A8,10I3,D22.15,D22.8)
 1100 FORMAT (2I5,2D10.1)
 1101 FORMAT(1X,'IOPT=',I2,5X,'MAXERR=',I4,5X,'SDMAX=',D11.3,5X,'FREQCY=
     *',F9.3/////)
 1102 FORMAT('1'///30X,'B I A S   V A L U E S'//
     314X,'STATION',13X,'PREPROCESSOR',12X,'NEW'/
     412X,'NAME   NUMBER',13X,'BIAS',16X,'BIAS'//)
 1006 FORMAT (///20X,'PROCESS COMPLETE')
      END
      SUBROUTINE UP601
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON  /U601/ PLOB,EMPLOB,ABIAS(60),XLABL,DATA(2),KAT,KONT,
     *KEY(10)
      DIMENSION IBIAS(60)
      DATA  IBIAS/957, 13,137,431,822, 54,766, 20,950,247, 22,876,
     *            78, 42,938,970,935,929,174,914, 72,975,872, 12,
     2            53,938, 94,870,12*0,
     3           948, 94, 83,889,959,807,842,926, 30,577, 76,126,
```

```
     4               64,949,603,  2, 31,935,950,120/
 1103 FORMAT(11X,A8,I5,14X,I4,16X,I4)
      READ(12) N,YLABL,DATA(1)
      IF(KEY(5).NE.10) RETURN
C
C     UPDATE BIAS
C
C     XK=0 OR 1 AND ULOBE=PLOB OR ELOB DEPENDING ON WHETHER THE STATION
C     NUMBER IS EVEN(POLAR) OR ODD(EQUATORIAL)
      IV=KEY(7)
      XK=KEY(2)-KEY(2)/2*2
      ULOBE=PLOB+EMPLOB*XK
      BIAS=DATA(1)/ULOBE
      LBIAS=BIAS+0.5D0
      IF(BIAS.LT.0.D0) LBIAS=BIAS-0.5D0
      BIAS=(BIAS-DFLOAT(LBIAS))
      DATA(1)= BIAS*ULOBE
      BIAS=BIAS*1000.D0 + DFLOAT(IBIAS(IV))
      ABIAS(IV)= DFLOAT(LBIAS)*ULOBE
      IF(BIAS.LT.-0.5D0) BIAS=BIAS+1000.D0
      IF(BIAS.GE.999.5D0) BIAS=BIAS-1000.D0
      LBIAS=BIAS+0.5D0
      WRITE (31,1103) XLABL,KEY(2),IBIAS(IV),LBIAS
      RETURN
      END
      SUBROUTINE IWRITE
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL*1 GRAPH,SYMBOL,STAR,ELEVOK,PLOT,BLANK,ELEV,WT
      DIMENSION SYMBOL(2),GRAPH(83),SIGMA(2),DMS(2),NOBS(2),PLOT(2),
     1 SUMT(2),SUMTSQ(2),SUMET(2),SCALE(2),TIMSEC(160),QMEAS(160,2),
     2 ERR(2),IGRAPH(2),KI(2),SLOPE(2),IROPE(2),NEDIT(2),STATIO(30),
     3 QMEAN(2),WT(2),KSEVEN(2)
      DATA BLANK,GRAPH /84*' '/
      DATA SYMBOL,STAR/'L','M','*'/
      DATA IGRAPH /21,63/
      DATA STATIO/'FTMYR6E ','FTMYR6P ','QUITO6E ','QUITO6P ',
     *            'LIMA6E ','LIMA6P ','SNTAG6E ','SNTAG6P ',
     *            'NEWFL6E ','NEWFL6P ','WNKFL6E ','WNKFL6P ',
     *            'JOBUR6E ','JOBUR6P ','ULASK6E ','ULASK6P ',
     *             'ORORA6E ','ORORA6P ','MADG6E ','MADGA6P ',
     *            'BPOIN6E ','BPOIN6P ','COLEG6E ','COLEG6P ',
     *            'MOJAV6E ','MOJAV6P ','GFORK6E ','GFORK6P ',
     *            'WOMER6E ','WOMER6P '  /
      DO 5 I=31,60
```

```
      M=I-30
      WRITE(I,1104) STATIO(M),M
    5 CONTINUE
   10 READ(10,END=200) NOARC,NOPASS,ISTSID,STLABL,ELEVOK,LMEAN,LA,SIGMA,
     *DMS,TDCURR,TSCURR,(TIMSEC(L),L=1,LA),((QMEAS(L,I),L=1,LA),I=1,2)
      READ(11) LF,IDAY,SEC,KEY1,KEY3,LERROR,MERROR,
     * LLOBE,MLOBE,LSD,MSD,KEY7L,KEY7M,KSEVEN
      IH= TSCURR/3600.D0
      SEC= TSCURR - (IH*3600)
      IM=SEC/60.D0
      SEC=SEC-(60*IM)
      DO 20 I=1,2
      PLOT(I) = STAR
      WT(I)=BLANK
      IF(KSEVEN(I).GT.0)  WT(I)=STAR
      IF(ELEVOK) PLOT(I)= SYMBOL(I)
      NOBS(I)=0
      SUMT(I)=0.D0
      SUMTSQ(I)=0.D0
      SUMET(I)=0.D0
      SCALE(I)= 0.15D0*SIGMA(I)
   20 CONTINUE
      IF(LA.LE.1) GO TO 45
      DO 28 I=1,2
      N=0
      SUM=0.D0
      TWOSIG=SIGMA(I)*3.D0
      DO 22 L=1,LA
      IF(QMEAS(L,I).GE.10.D20) GO TO 22
      SUM=SUM+QMEAS(L,I)
      N=N+1
   22 CONTINUE
      N1=N
      SUM1=SUM
   24 CONTINUE
      NOLD=N
      EN=N
      AVRAGE=SUM/EN
      N=N1
      SUM=SUM1
      DO 26 L=1,LA
      IF(QMEAS(L,I).GE.10.D20) GO TO 26
      IF(DABS(QMEAS(L,I)-AVRAGE).LE.TWOSIG) GO TO 26
      SUM = SUM-QMEAS(L,I)
```

```
      N=N-1
   26 CONTINUE
      IF(N.LT.NOLD) GO TO 24
      QMEAN(I)= AVRAGE
   28 CONTINUE
      WRITE(20,1105) STLABL,ISTSID,NOARC,NOPASS,IDAY,IH,IM,SEC,QMEAN,
     *SIGMA
      DO 40 L= 1,LA
      TIM= TIMSEC(L)- TSCURR
      IF(TIM.GT.3600.D0) TIM= TIM- 86400.D0
      IF(TIM.LT.-3600.D0) TIM = TIM + 86400.D0
      DO 30 I=1,2
      ERR(I)= QMEAS(L,I)-QMEAN(I)
      IF(ERR(I).GE.10.D20) GO TO 500
      ERRORP= ERR(I)/SCALE(I)
      IF (DABS(ERRORP).GT.20.D0) GO TO 520
      KI(I) = ERRORP + IGRAPH(I)
      GRAPH(KI(I)) = PLOT(I)
      NOBS(I)= NOBS(I)+1
      SUMT(I)= SUMT(I)+ TIM
      SUMTSQ(I) = SUMTSQ(I) + TIM*TIM
      SUMET(I)= SUMET(I)+ ERR(I)*TIM
   30 CONTINUE
      WRITE(20,1106) TIM,ERR,GRAPH
      GRAPH(KI(1)) = BLANK
      GRAPH(KI(2)) = BLANK
   40 CONTINUE
   45 CONTINUE
      DO 50 I=1,2
      PLOT(I)= BLANK
      IF(LA.LE.1) NOBS(I)=LA
      NEDIT(I)= LA - NOBS(I)
      SLOPE(I)= 0.D0
      IROPE(I)=0
      IF(NOBS(I).LE.2) GO TO 50
      OBSNUM= NOBS(I)
      TBAR= SUMT(I)/OBSNUM
      SLOPE(I)= SUMET(I)/(SUMTSQ(I)-SUMT(I)*TBAR)
      IROPE(I)=100
      TP=1.D0+(1.D0-SLOPE(I)*SUMET(I)/(SIGMA(I)*SIGMA(I)))/(OBSNUM-2.D0)
      IF(TP.GT.0.D0) IROPE(I)=100.D0*(1.005D0-DSQRT(TP))
   50 CONTINUE
      ELEV=STAR
      IF(ELEVOK) ELEV=BLANK
```

A-2.8 SAMPLE JCL FOR COMPLETE NAP RUN

JCL FOR COMPLETE NAP RUN

```
//******************************(JOB CARD)********************************
      JCL FOR PRENAP:

// EXEC LOADER,PARM='MAP,CALL,SIZE=100K',REGION.GO=110K
//GO.SYSLIN DD *

      *****OBJECT DECK FOR PRENAP-CARD-UPDATE PROGRAM INSERTED HERE*****

/*
//GO.FT09F001 DD UNIT=9TRACK,LABEL=(11,BLP),DSN=ITAPEGEM,
// DCB=(RECFM=VBS,LRECL=80,BLKSIZE=3204),DISP=(OLD,PASS),
// VOL=SER=34503G
//GO.FT10F001 DD UNIT=DISK,SPACE=(TRK,(1,1)),
// DCB=(RECFM=VBS,LRECL=84,BLKSIZE=172),DISP=(NEW,DELETE)
//GO.FT12F001 DD UNIT=DISK,SPACE=(CYL,(3,1)),DSN=NAPOLEON,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),DISP=(NEW,PASS)
//GO.DATA5 DD *
MOD     1        1
              1
MOD     9        9
                                             .72000000000000000D04.5000000000D-13
REP    141      152
601 OFTMYRELR  0  1  0  1 15  1 29  1  0  0.0                   0.1D-2
601 OFTMYRPMR  0  2  0  1 15  4 30  1  0  0.0                   0.1D-2
601 OQUITCELR  0  3  0  1 15  5 31  1  0  0.0                   0.1D-2
601 OQUITOPMR  0  4  0  1 15  8 32  1  0  0.0                   0.1D-2
601 OLIMAPELR  0  5  0  1 15  9 33  1  0  0.0                   0.1D-2
601 OLIMAPPMR  0  6  0  1 15 12 34  1  0  0.0                   0.1D-2
601 OSNTAGELR  0  7  0  1 15 13 35  1  0  0.0                   0.1D-2
601 OSNTAGPMR  0  8  0  1 15 16 36  1  0  0.0                   0.1D-2
601 ONEWFLELR  0  9  0  1 15 17 37  1  0  0.0                   0.1D-2
601 ONEWFLPMR  0 10  0  1 15 20 38  1  0  0.0                   0.1D-2
601 OWNKFLELR  0 11  0  1 15 21 39  1  0  0.0                   0.1D-2
601 OWNKFLPMR  0 12  0  1 15 24 40  1  0  0.0                   0.1D-2

MOD    173      186
                                             0.0000000000000000000D00.1000000000D-02
REP    187      192
601 OJOBURELR  0 13  0  1 15 25 75  1  0  0.0                   0.1D-2
601 OJOBURPMR  0 14  0  1 15 28 76  1  0  0.0                   0.1D-2
601 OBPOINELR  0 21  0  1 15 41 77  1  0  0.0                   0.1D-2
601 OBPOINPMR  0 22  0  1 15 44 78  1  0  0.0                   0.1D-2
601 OCOLEGELR  0 23  0  1 15 45 79  1  0  0.0                   0.1D-2
```

```
601 OCOLEGPMR  0 24  0  1 15 48 80  1  0   0.0                                     0.1D-2

MOD    193         202
                                          0.0000000000000000000D00.1000000000D-02

INS    202         202
601 OMOJAVELR  0 25  0  1 15 49 91  1  0   0.0                                     0.1D-2
601 OMOJAVPMR  0 26  0  1 15 52 92  1  0   0.0                                     0.1D-2
601 OGFORKELR  0 27  0  1 15 53 93  1  0   0.0                                     0.1D-2
601 OGFORKPMP  0 28  0  1 15 56 94  1  0   0.0                                     0.1D-2
601 OOOMERELR  0 29  0  1 15 57 95  1  0   0.0                                     0.1D-2
601 OOOMERPMR  0 30  0  1 15 60 96  1  0   0.0                                     0.1D-2

MOD    1403        1403
                          0                                           0.7000000000D-03
MOD    1436        1436
                                          .7200000000000000000D04.5000000000D-13
MOD    2523        2523
                                          .7200000000000000000D04.5000000000D-13
MOD    2763        2763
                          0                                           0.5000000000D-3
MOD    2802        2802
                          0                                           0.7000000000D-03
MOD    3177        3177
                          0                                           0.3000000000D-03
MOD    3297        3297
                          0                                           0.3500000000D-02
MOD    3443        3443
                          0                                           0.7470000000D-03
MOD    3708        3708
                          0                                           0.8650000000D-03
MOD    4087        4087
                          0                                           0.3610000000D-03
MOD    4230        4230
                                          .7200000000000000000D04.5000000000D-13
MOD    5165        5165
                          0                                           0.3000000000D-02
MOD    5400        5400
                          0                                           0.2810000000D-02
MOD    5515        5515
                          0                                           0.6000000000D-02
MOD    5555        5555
                          0                                           0.8000000000D-03
/*
```

JCL FOR NAP:

```
//  EXEC  LINKGO,PARM='OVLY,MAP,LIST,XREF',NBLK=300,REGION.GO=499K
//LINK.SYSPRINT DD SYSOUT=A,SPACE=(CYL,(2,1))
//LINK.SYSUT1 DD SPACE=(CYL,(5,2))
//LINK.TAPELIB DD UNIT=2400-9,LABEL=(2,BLP,,IN),DISP=(OLD,PASS),
//  DSN=NAPOBJ,
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),VOL=SER=34503G
//LINK.SYSLIN  DD *
 INSERT  CLEAN
 INSERT  COMSOL,TSPARM,ASPARM,ACINFO,PSPARM,STINFO,TYLE,GENCOM,MSINFO
 INSERT CDEBUG,IONUMB,GPCOM,EXTCM,EARTH,ICONST,FCONST,CONMET
 INSERT  CINTEG,CMEASR,RSUMR
 INSERT  MAIN,PAGE,ROTFIX,ROTINT,RESID,SIGWT,DAYHMS,CWORK
 INSERT  CLEAR
 OVERLAY ALPHA
 INSERT  DREDIT,SPOLCD,DEFALT,INPCRD,GENFIL,EDIT,GAUSS,RANDU
 INSERT  UNIFD2,GPRSM2,GEOS,EDPOLY,REPRT2,RDMS,COMGHA,JULDAY,ECINT
 OVERLAY ALPHA
 INSERT  PRTIAL
 INSERT MATMUT
 INSERT  REFRCT,ROTPAR,MEASUR,RSUM,MESOLD,ROTBAK,ROTOLO,ROTOMA,AZIM
 INSERT  ELEV,RARIES,SPOVEL,SVAREQ
 INSERT  SOFORT,SOFSEC,KICKER,ENGRAT
 INSERT  EXPAND
 INSERT SOLVEX,SOVARY,ENTRNS,SOLANG,ENROOT,CLEAR
 INSERT CKTIME,ENUTAT,PENMAT,PRENUT,VARIEQ,ENEXPS
 INSERT  ENVARY,SUMBDY,ENDIST,KCOM,JCOM,ICOM,HCOM,GCOM,FCOM,ECOM
 INSERT  DCOM,CCOM,BCOM,ACOM,PRIMY
 OVERLAY ALPHA
 INSERT  SOLVER,INVERT,FINALP
 OVERLAY ALPHA
 INSERT  SIMOUT

   *********ANY OBJECT DECKS FOR LINK STEP ARE INSERTED HERE*********

 INCLUDE TAPELIB
 ENTRY MAIN
//LINK.SYSUDUMP DD SYSOUT=A,SPACE=(CYL,(1))
//GO.FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
//  SPACE=(CYL,(9,1))
//SYSPRINT DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265)
//GO.FT10F001 DD UNIT=DISK,SPACE=(TRK,(3,1)),DSN=ATGOUTP,
```

```
//     DCB=(RECFM=VBS,LRECL=64,BLKSIZE=644),DISP=(NEW,PASS)
//GO.FT21F001 DD UNIT=DISK,SPACE=(CYL,(10,1)),DSN=Z7GEM21,
//         DCB=(RECFM=VBS,LRECL=156,BLKSIZE=7180)
//GO.FT22F001 DD UNIT=DISK,SPACE=(CYL,(4,1)),
//         DCB=(RECFM=VBS,LRECL=304,BLKSIZE=6996)
//GO.FT23F001 DD UNIT=2314,SPACE=(TRK,(4,1)),
//  DCB=(RECFM=VBS,LRECL=200,BLKSIZE=404)
//GO.FT24F001 DD UNIT=2314,SPACE=(TRK,(8,1)),
//  DCB=(RECFM=VBS,LRECL=200,BLKSIZE=404)
//GO.FT25F001 DD UNIT=DISK,SPACE=(CYL,(1,1)),
//  DCB=(RECFM=VBS,LRECL=200,BLKSIZE=404)
//GO.FT26F001 DD UNIT=2314,SPACE=(TRK,(8,1)),
//  DCB=(RECFM=VBS,LRECL=200,BLKSIZE=404)
//GO.FT27F001 DD UNIT=DISK,SPACE=(CYL,(3,1)),DSN=ATGCARDS,
//      DCB=(RECFM=VBS,LRECL=88,BLKSIZE=884),DISP=(NEW,DELETE)
//GO.FT28F001  DD  DUMMY
//GO.FT29F001 DD UNIT=2400-9,LABEL=(5,BLP),DISP=(OLD,PASS),
//    DCB=(RECFM=VBS,LRECL=52,BLKSIZE=5204),VOL=SER=34503G
//GO.FT30F001 DD DUMMY
//GO.FT31F001 DD UNIT=2314,SPACE=(TRK,(4,1)),
//  DCB=(RECFM=VBS,LRECL=200,BLKSIZE=404)
//GO.FT32F001 DD UNIT=DISK,SPACE=(CYL,(9,1)),DSN=Z7GEM32,
//      DCB=(RECFM=VBS,LRECL=2168,BLKSIZE=2172)
//GO.FT33F001 DD DUMMY
//GO.FT35F001 DD DUMMY
//GO.FT36F001 DD UNIT=DISK,SPACE=(CYL,(1,1)),DSN=Z7GEM36,
// DCB=(RECFM=VBS,LRECL=36,BLKSIZE=364),DISP=(NEW,DELETE)
//GO.FT37F001 DD UNIT=AFF=FT29F001,LABEL=(13,BLP),DSN=NURITE,
// DCB=(RECFM=VBS,LRECL=3825,BLKSIZE=7654),DISP=(NEW,PASS),
// VOL=SER=34503G
//GO.FT37F002 DD UNIT=AFF=FT29F001,LABEL=(14,BLP),DSN=NUPASS,
// DCB=(RECFM=VBS,LRECL=64,BLKSIZE=644),DISP=(NEW,PASS),
// VOL=SER=34503G
//GO.FT37F003 DD DUMMY
//GO.SYSABEND DD SYSOUT=A,SPACE=(CYL,(1))
//GO.DATA5 DD UNIT=DISK,DSN=NAPOLEON,DISP=(OLD,PASS)
/*
```

JCL FOR POST_NAP:

```
//    EXEC  LOADER,PARM='MAP,CALL,SIZE=180K',REGION.GO=190K
//GO.SYSLIN DD *
```

```
**********************OBJECT DECK INSERTED HERE**********************
/*
//GO.FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
//          SPACE=(CYL,(5,1))
//GO.FT08FC01 DD UNIT=DISK,SPACE=(CYL,(4,1)),DSN=Z7GEM08,
// DCB=(RECFM=VBS,LRECL=80,BLKSIZE=1604),DISP=(NEW,DELETE)
//GO.FT10F001 DD UNIT=9TRACK,LABEL=(14,BLP),DSN=Z7GEM10,
// DCB=(RECFM=VBS,LRECL=64,BLKSIZE=3204),DISP=(OLD,PASS),
// VOL=SER=34503G
//GO.FT10F002 DD UNIT=9TRACK,LABEL=(13,BLP),DSN=Z7GEM1A,
// DCB=(RECFM=VBS,LRECL=3825,BLKSIZE=7654),DISP=(OLD,PASS),
// VOL=SER=34503G
//GO.FT10F003 DD UNIT=9TRACK,LABEL=(15,BLP),DSN=Z7GEM1B,
// DCB=(RECFM=VBS,LRECL=80,BLKSIZE=1604),DISP=(NEW,KEEP),
// VOL=SER=34503G
//GO.FT11F001 DD UNIT=DISK,SPACE=(CYL,(3,1)),DSN=Z7GEM11,
// DCB=(RECFM=VBS,LRECL=76,BLKSIZE=764),DISP=(NEW,DELETE)
//GO.FT12FC01 DD UNIT=DISK,SPACE=(TRK,(1,1)),DSN=Z7GEM12,
// DCB=(RECFM=VBS,LRECL=24,BLKSIZE=244),DISP=(NEW,DELETE)
//GO.FT20F001 DD DUMMY
//GO.FT27FC01 DD UNIT=DISK,DSN=NAPOLEON,DISP=(OLD,DELETE)
//GO.FT31F001 DD SYSOUT=A,SPACE=(TRK,(4,1)),
//  DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT32FC01 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//  DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT33F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//  DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT34FC01 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//  DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT35FC01 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//  DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT36F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//  DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT37FC01 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//  DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT38FC01 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//  DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT39F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//  DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT40FC01 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//  DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT41F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//  DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
```

```
//GO.FT42F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT43F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT44F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT45F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT46F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT47F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT48F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT49F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT50F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT51F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT52F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT53F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT54F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT55F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT56F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT57F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT58F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT59F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.FT60F001 DD SYSOUT=A,SPACE=(TRK,(2,1)),
//   DCB=(RECFM=VBA,LRECL=137,BLKSIZE=689)
//GO.DATA5 DD *
   1   20    0.5D-3  136.830
/*
```

0261 CARDS

APPENDIX A-3

SUBROUTINE MODIFICATIONS TO NAP-II

A-3.1 SUBROUTINE ENEXPS

C.5

```
      SUBROUTINE ENEXPS(ITN)
C
C     SUBROUNTINE ENEXPS(ITN) COMMENTS
C
C     1.  FOR EACH POWER SERIES COMPUTED THIS SUBROUTINE IS CALLED
C     (KTR-2  )   TIMES BY SUBROUTINE EXPAND.  KTR IS THE NUMBER OF
C     TERMS IN THE POWER SERIES FOR POSITION (X,Y,Z) AND ITN IS THE
C     TERM NUMBER BEING COMPUTED ON THE CURRENT ITERATION.
C     INITIAL POSITION AND VELOCITY WOULD CORRESPOND TO ITN=1 AND 2
C     RESPECTIVELY AND ARE NOT COMPUTED BY THIS SUBROUTINE.  FOR EACH
C     POWER SERIES THAT THIS SUBROUTINE IS CALLED THE INITIAL VALUE
C     OF ITN IS THUS 3.
C
C
C     2.  THIS SUBROUTINE COMPUTES THE EFFECTS OF THE GRAVITATIONAL
C     FORCE OF N-BODIES ON EACH OTHER. THE EFFECT OF SOLAR RADIATION
C     PRESSURE IS ALSO COMPUTED.  HOWEVER, THE GRAVITATIONAL PULL
C     OF THE PRIMARY SOURCE ON  THE PROBE IS NOT COMPUTED HERE BUT
C     IN SUBROUTINE EXPAND.
C
C
C     3.  DENOTING (ITN-3) BY K, THE COEFFICIENTS OF T**K FOR RELATIVE
C     DISTANCES AND THE COEFFICIENTS T**(K+2) FOR POSITIONS RELATIVE
C     TO THE PRIMARY SOURCE ARE COMPUTED.  NOTE THAT THE COEFFICIENT OF
C     T**(K+2) IS ACTUALLY THE (3+K)TH, I.E. ITN-TH COEFFICIENT.  THE
C     CONTRIBUTION TO  THE ITN-TH POSITION COEFFICIENTS OF THE PROBE
C     OF THE PRIMARY SOURCE GRAVITY FIELD AND DRAG ARE CONTAINED IN
C     (XPO,YPO,ZPO) WHEN THIS ROUTINE IS CALLED.
C
C
C     4.  THE COEFFICIENTS OF THE RELATIVE BODY POSITIONS ARE ARRANGED
C     SUCH THAT SEQUENTIAL COEFFICIENTS OF ANY ONE BODY ARE ALWAYS
C     'NBD' COEFFICIENTS APART  'NBD' THUS APPEARS AS A UNIT AND HAS
C     FOR THIS REASON BEEN EQUIVALENCED TO 'ONE'.  THE UNIT 'UNIT'
C     CORRESDONDING TO RELATIVE DISTANCES IS NBD(NBD-1)/2, THE NUMBER OF
C     RELATIVE DISTANCES BETWEEN NBD BODIES.  THE VALUE OF K APPEARING
C     IN THE PROGRAM IS GIVEN BY (ITN-3)* ONE AND CORRESPONDS THE
C     K OF COMMENT 3 ABOVE.  THUS X(N+K) IS THE COEFFICIENT OF T**K OF
C     X-POSITION OF BODY N.
C
C
C     5.  SINCE ITN IS INITIALLY 3 (SEE COMMENT 1) IT FOLLOWS THAT K
C     IS INITIALLY 0.
C
```

```
C
C
C
C     6A.  IF SOLAR PRESSURE IS NOT BEING COMPUTED (KSP=0) THEN NO
C     COMPUTATIONS ARE PERFORMED FOR THE PROBE-PRIMARY SOURCE
C     COMBINATION (SEE COMMENT 2).
C
C        IF SOLAR PRESSURE IS BEING COMPUTED THEN THE PROBE-PRIMARY
C     SOURCE DISTANCE MUST BE COMPUTED (BUT NOT THE PRIMARY SOURCE
C      GRAVITATIONAL FORCE).
C     6B.  IF THE SUN IS NOT THE PRIMARY SOURCE, THEN SHADOW
C     COMPUTATIONS ARE PERFORMED, BUT
C
C     6C.   IF THE SUN IS THE PRIMARY SOURCE, THEN NO SHADOW
C     COMPUTATIONS ARE PERFORMED.
C
C
C       DENOTING THE PROBE-PRIMARY SOURCE POSITION IN THE R, THETA
C     AND PHI ARRAYS BY N (WHERE N.LE.45) THE FOLLOWING QUANTITIES ARE
C     COMPUTED
C     FOR CASE 6A. (NPRNPS=   N, NPSNPR= 100, KSP2= 0)
C     FOR CASE 6B. (NPRNPS= 100, NPSNPR=   N, KSP2= 1)
C     FOR CASE 6C. (NPRNPS= 100, NPSNPR= 100, KSP2= 0)
C
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*4 ONE,TWO,UNIT,S
      REAL*8 KP1KP2
      COMMON /CCOM/
     1XPO(16),YPO(16),ZPO(16),XBD(160),YBD(160),ZBD(160),CGB(14),SGB(14)
     2,XTL(14),YTL(14),CLB(14),SLB(14),CLT(14),SLT(14),VRB(14),
     3RPT(14),RMT(14),RMO(14),RZR(14),RMR(14)
      COMMON /ECOM/
     1BDY(10),BMU(10),BTM(10),HTL(16),BRT(10),BRA(10),BDT(10),STL(16)
     2,RST(825),VST(825),WST(770),NCN(10),NBT(10)
      COMMON /ICOM/
     1NBD,NTE,NHT,NSU,NPR,NPS,NOS,ICN,KTR,KDR,KVE,KDV,KSP,ISP,KIN,MPT
      EQUIVALENCE (A(1),STL(1)),(THETAC(1),HTL(1)),(X(1),XBD(1)),
     *  (Y(1),YBD(1)),(Z(1),ZBD(1)),(R(1),RST(1)),(THETA(1),VST(1)),
     *  (PHI(1),WST(1)),(ONE,NBD)
      DIMENSION A(16),THETAC(16),X(160),Y(160),Z(160),R(825),THETA(825),
     *  PHI(770)
      IJO= 0
```

```
      XK= ITN - 3
      QK3= -XK/3.DO
      KP1KP2= (XK+1.DO)*(XK+2.DO)
      K= (ITN-3) * ONE
C
C     FOR (K=0) SEE COMMENT 5.
      IF(K.LE.0) GO TO 1300
C
  100 KMONE= K - ONE
      KPTWO= K + TWO
C
      DO 700 IO= 1,NBDM1
      IK= IO + K
      IKPTWO= IK + TWO
      GMI= BMU(IO)
C
      IOP1=IO+1
      DO 600 JO=IOP1,NBD
      JK =  JO + K
      IJO= IJO + 1
      IJS= IJO
      IJK= IJK + 1
      IJKMS= IJK
C
C     FOR 'IJO=NPRNPS' SEE COMMENT 6.
C
      IF(IJO.EQ.NPRNPS) GO TO 600
C
      XIJO= X(IO) - X(JO)
      YIJO= Y(IO) - Y(JO)
      ZIJO= Z(IO) - Z(JO)
C
C     FOR 'K=0' SEE COMMENT 5.
C
      IF(K.LE.0) GO TO 1200
C
      FXIJK= THETA(IJO) * (X(IK)-X(JK))
      FYIJK= THETA(IJO) * (Y(IK)-Y(JK))
      FZIJK= THETA(IJO) * (Z(IK)-Z(JK))
      RIJK= XIJO*(X(IK)-X(JK)) + YIJO*(Y(IK)-Y(JK)) + ZIJO*(Z(IK)-Z(JK))
      THIJK= 0.DO
      PHIJK= 0.DO
C
      IF(KMONE.LE.0) GO TO 400
```

```
C
      DO 300 S= ONE,KMONE,ONE
      IJS  = IJS   + UNIT
      IJKMS= IJKMS - UNIT
      PHIJK=      R(IJS) * PHI(IJKMS) + PHIJK
      THIJK= THETA(IJS) * PHI(IJKMS) + THIJK
      FXIJK= THETA(IJS) * (X(IK-S) - X(JK-S)) + FXIJK
      FYIJK= THETA(IJS) * (Y(IK-S) - Y(JK-S)) + FYIJK
      FZIJK= THETA(IJS) * (Z(IK-S) - Z(JK-S)) + FZIJK
      IF(IJS-IJKMS) 200,1000,300
C
  200 RIJK= (X(IO+S)-X(JO+S)) * (X(IK-S)-X(JK-S))
     1     + (Y(IO+S)-Y(JO+S)) * (Y(IK-S)-Y(JK-S))
     2     + (Z(IO+S)-Z(JO+S)) * (Z(IK-S)-Z(JK-S))
     3     - R(IJS) * R(IJKMS) + RIJK
  300 CONTINUE
C
  400 R(IJK)     = RIJK/R(IJO)
C
C     FOR 'IJO=NPSNPR' SEE COMMENT 6.
      IF(IJO.EQ.NPSNPR) GO TO 2000
C
      PHI(IJK)    = (R(IJK)*XK - PHIJK)/R(IJO)
      THETA(IJK) = (PHI(IJK)*THETA(IJO) + THIJK)/QK3
      FXIJK = THETA(IJK)*XIJO + FXIJK
      FYIJK = THETA(IJK)*YIJO + FYIJK
      FZIJK = THETA(IJK)*ZIJO + FZIJK
C
C
  500 IF(IJO.EQ.NPRNSU) GO TO 1100
C
      X(IKPTWO) = - BMU(JO)*FXIJK + X(IKPTWO)
      Y(IKPTWO) = - BMU(JO)*FYIJK + Y(IKPTWO)
      Z(IKPTWO) = - BMU(JO)*FZIJK + Z(IKPTWO)
C
      X(JK+TWO) =          GMI*FXIJK + X(JK+TWO)
      Y(JK+TWO) =          GMI*FYIJK + Y(JK+TWO)
      Z(JK+TWO) =          GMI*FZIJK + Z(JK+TWO)
C
  600 CONTINUE
  700 CONTINUE
C
  800 L1= KPTWO + 1
      L2= L1 + NBDM1
```

```
C
      FXIJK= X(NPS+KPTWO)
      FYIJK= Y(NPS+KPTWO)
      FZIJK= Z(NPS+KPTWO)
      DO 900 L= L1,L2
      X(L)= (X(L) - FXIJK)/KP1KP2
      Y(L)= (Y(L) - FYIJK)/KP1KP2
      Z(L)= (Z(L) - FZIJK)/KP1KP2
  900 CONTINUE
C
C     SEE COMMENT 3.
C
C
      X(NPR+KPTWO) = X(NPR+KPTWO) + XPO(ITN)
      XPO(ITN)     = X(NPR+KPTWO)
      Y(NPR+KPTWO) = Y(NPR+KPTWO) + YPO(ITN)
      YPO(ITN)     = Y(NPR+KPTWO)
      Z(NPR+KPTWO) = Z(NPR+KPTWO) + ZPO(ITN)
      ZPO(ITN)     = Z(NPR+KPTWO)
C
C     KSP2 IS GREATER THAN ZERO IF SHADOW COMPUTATIONS ARE REQUIRED.
C
      IF(KSP2.GT.0) GO TO 2300
C
      RETURN
C
C
C
C
 1000 RIJK= ((X(IK-S) -X(JK-S))**2 + (Y(IK-S) - Y(JK-S))**2
     *     + (Z(IK-S) -Z(JK-S))**2 - R(IJS)**2) * 0.5D0 + RIJK
      GO TO 300
C
C
 1100 X(NPR+KPTWO) = GMSUN * FXIJK + X(NPR+KPTWO)
      Y(NPR+KPTWO) = GMSUN * FYIJK + Y(NPR+KPTWO)
      Z(NPR+KPTWO) = GMSUN * FZIJK + Z(NPR+KPTWO)
      GO TO 600
C
C
 1200 RIJK = XIJO**2 + YIJO**2 + ZIJO**2
      R(IJO) = DSQRT(RIJK)
      THETA(IJO) = R(IJU)/RIJK/RIJK
      FXIJK = THETA(IJO) * XIJO
```

```
      FYIJK = THETA(IJO) * YIJO
      FZIJK = THETA(IJO) * ZIJO
      IF (IJO .EQ. NPSNPR) GO TO 600
      GO TO 500
C
C
C
C
 1300 NBDM1= NBD - 1
      UNIT= NBDM1*NBD/2
      TWO= ONE + ONE
      L1 = TWO + 1
      L2 = KTR*ONE
C
      DO 1400 L= L1,L2
      X(L)= 0.D0
      Y(L)= 0.D0
      Z(L)= 0.D0
 1400 CONTINUE
C
      ITNM2=1
      IJK= 0
      MNSU= NBD*(NSU-1) - NSU*(NSU+1)/2
      MNPR= NBD*(NPR-1) - NPR*(NPR+1)/2
      MNPS= NBD*(NPS-1) - NPS*(NPS+1)/2
      NPRNSU= MNPR + NSU
      IF(NSU.LT.NPR) NPRNSU= MNSU + NPR
      NPRNPS= MNPR + NPS
      IF(NPS.LT.NPR) NPRNPS= MNPS + NPR
      GMSUN= BMU(NSU)
      IF(KSP.GT.0) GO TO 1500
 1440 NPSNPR= 100
      KSP2= 0
 1450 IF(NPR.LT.NSU) GMSUN= -GMSUN
      GO TO 100
C
 1500 NPSNPR= NPRNPS
      NPRNPS= 100
      IF(NPS.EQ.NSU) GO TO 1550
      NSUNPS= MNPS + NSU
      IF(NSU.LT.NPS) NSUNPS= MNSU + NPS
      KSP2 = 1
      A(KTR)       = 0.D0
      THETAC(KTR)  = 0.D0
```

```
      A(KTR-1)      = 0.D0
      THETAC(KTR-1)= 0.D0
      IF(ISP.GT.0) GMSUN= GMSUN - SPC
      GO TO 1450
C
C
 1550 GMSUN= -SPC
      GO TO 1440
C
C
C
C     SOLAR PRESSURE SHADOW SERIES.
C
 2000 AK = R(IJK)* R(IJO)
      ITNM2 = ITN - 2
      L = ITNM2/2
C
C
C     NOTE THAT , PROVIDED THAT K IS GREATER THAN ZERO, THEN AT THE END
C     OF DO-LOOP 300   IJS = (K-1) UNITS  AND  IJKMS = 1 UNIT.
C     L IS CHOSEN SUCH THAT IF ITN IS EVEN THEN  L = (ITN-1-L)-1  AND
C     IF ITN IS ODD THEN   L = (ITN-1-L)-2 .  IN THE LATTER CASE AN
C     EXTRA TERM MUST BE ADDED TO  AK.  THE LATTER CONDITION IS
C     EQUIVALENT TO  IJKMS = IJS.
C
C
      IF(L.LE.1) GO TO 2200
      DO 2100 S = 2,L
      AK= R(IJKMS)*R(IJS) - A(S)*A(ITN-1-S) + AK
      IJKMS= IJKMS + UNIT
      IJS  = IJS   - UNIT
 2100 CONTINUE
 2200 IF(IJKMS.LE.IJS) AK= (R(IJS)**2 - A(L+1)**2) * 0.5D0 + AK
      A(ITNM2)= AK/A(1)
      GO TO 600
C
C
C
C
C     IJK AT THIS POINT EQUALS (K+1) UNITS.
C
 2300 THETAK= 0.D0
      DO 2400 S= 1,ITNM2
      THETAK= X(NSU-ONE+S*ONE) * X(NPR+K+ONE-S*ONE)
```

```
     1       + Y(NSU-ONE+S*ONE) * Y(NPR+K+ONE-S*ONE)                    104510
     2       + Z(NSU-ONE+S*ONE) * Z(NPR+K+ONE-S*ONE)                    104520
     3       + A(S)*R(NSUNPS+IJK-S*UNIT) + THETAK                       104530
 2400 CONTINUE                                                          104540
      THETAC(ITNM2)= THETAK                                             104550
      RETURN                                                            104560
C                                                                       104570
      END                                                               104580
                                                                            0316 CARDS
```

A-3.2 SUBROUTINE EXPAND

```
      SUBROUTINE EXPAND (KEY,NNA)
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 ONE
      DIMENSION IDUMMY(2),U(2394),V(2394)
C     COMMON BLOCK - REDUCED FORM
      COMMON /ACOM/
     1CNM(136),SNM(136),F(171),C00,C20,EXTRA(99),UMT(2394),VMT(2394)
     2,LCT(16),ICT(17),ONE
      COMMON /CCOM/
     1XPO(16),YPO(16),ZPO(16),XBD(160),YBD(160),ZBD(160),CGB(14),SGB(14)
     2,XTL(14),YTL(14),CLB(14),SLB(14),CLT(14),SLT(14),VRB(14),
     3RPT(14),RMT(14),RMO(14),RZR(14),RMR(14)
      COMMON /DCOM/
     1HTS(12),CAV(12),CBV(12),ALT(12),CNA(12),CNB(12),XRD(14),YRD(14),
     2ZRD(14),XDD(14),YDD(14),ZDD(14),RHO(14),RPO(14),RVR(14),HNV(14),
     3VRN(14),VSR(14)
      COMMON /HCOM/
     1TON,TEN,EPS,TIN,DEL,SPC,PSA,FSA,TOJ,UPD,TOP,DTP
     2,CUD,CUV,CUT,ERD,XMU,ALF,OMG,ECC,CDC,CTW,XIT(6)
      COMMON /ICOM/
     1NBD,NTE,NHT,NSU,NPR,NPS,NOS,ICN,KTR,KDR,KVE,KDV,KSP,ISP,KIN,MPT
C     COMMON BLOCK COMPLETED
      EQUIVALENCE (IDUMMY(1),I,XXXX),(IDUMMY(2),J2),
     *(U(1),UMT(1)),(V(1),VMT(1))
      EQUIVALENCE (ONE,J3)
      SIN(XXX) = DSIN(XXX)
      COS(XXX) = DCOS(XXX)
      EXP(XXX) = DEXP(XXX)
      ATAN(XXX) = DATAN(XXX)
      SQRT(XXX) = DSQRT(XXX)
       IF (KEY . GT. 2) GO TO 100
       XMUE = XMU /(ERD +ERD)
       J = 1
       DO 30 M = 1,NTE
       NM = LCT (M)
       IF(NM . LE. 0) GO TO 30
      XMN=XMUE
       I2MP1 = M+M-1
      XMNO= I2MP1
       XDIV = 0.D0
       DO 10 I = 1, I2MP1,2
      X=I
   10 XMN=XMN*X
       DO 20 K=1,NM
```

```
      CNM (J) = CNM (J)*XMN
      SNM (J) = SNM(J) * XMN
      J= J+1
      XMNO = XMNO +  2.DO
      XDIV = XDIV +1.DO
20    XMN = XMN * XMNO/XDIV
30    CONTINUE
      J1 = 1
      NA = NTE +1
      KMAZ = 0
      LA = 0
      LB = LCT (1)
      IF(KVE.LE.2) GO TO 50
      C00= 6.DO*CNM(1)/ERD
      C20= 14.DO*CNM(3)/ERD
      IF(LB.LE.2) C20= 0.DO
      IF ( NA . GE. 3 ) GO TO 40
      NA = 3
      LCT (2) = 0
40    KMAZ = 3
      IF (LB . GT. 2 ) KMAZ = 5
50    CONTINUE
      DO 90  MP1 = 1,NA
      IF (LA . LT. (KMAZ-MP1)) LA = KMAZ-MP1
      IF (LA . LT. LB) LA = LB
      IF (MP1 . GE. NA ) GO TO 60
      LC = LCT (MP1+1)
      IF (LC . LE. 0 ) GO TO 60
      IF ( LA . LE. LC )  LA = LC +1
60    J3 = J1 + LA
      J2 = J3 -1
      IF ( LA . LE. 0 ) GO TO 80
      JP1 = J1 + 1
      XMSQ = (MP1-1) * (MP1-1)
      XN = MP1
      DO  70 J = JP1,J3
      F(J) = (XMSQ - XN**2)/(4.DO*XN**2 - 1.DO )
70    XN = XN + 1.DO
80    I = 1 - LA
      IF ( I . LT. 0 )  I = -1
      IF (MP1 .GT. 1) I = I + 2
      IF ( MP1. GE. NA ) I= I + 3
      F(J1) = XXXX
      J1 = J3 + 1
```

```
      ICT(MP1)= J3
      LA = LB - 1
      LB = LC
   90 CONTINUE
      KEY = 3
  100 CONTINUE
     ITR=3
      K = 1
      KO=0
     KJ=1
     CGB(1)=DCOS(ALF)
     SGB(1)=DSIN(ALF)
     CLB(1)=XPO(1)*CGB(1)+YPO(1)*SGB(1)
     SLB(1)=YPO(1)*CGB(1)-XPO(1)*SGB(1)
     RPT(1)=XPO(1)*XPO(1)+YPO(1)*YPO(1)+ZPO(1)*ZPO(1)
     RMT(1)=1.DO/RPT(1)
     CLT(1)=ERD*RMT(1)*CLB(1)
     SLT(1)=ERD*RMT(1)*SLB(1)
     RZR(1)=ERD*RMT(1)*ZPO(1)
     RMR(1)=ERD*ERD*RMT(1)
     RMO(1)=DSQRT(RMT(1))
     RMOTWO= RMO(1)+RMO(1)
     U(1)= RMO(1)
      GO TO 490
  300 CONTINUE
     TX=0.DO
     TY=0.DO
     TZ=0.DO
     KONE= KJ-ONE
     IF(NTE.LE.1) GO TO 340
     KO= LCT(1)
     L= ICT(1) + 1 + KONE
     LM1= 2 + KONE
     LP1=ICT(2)+ KONE
     IN= LCT(2)
     IF(IN.LE.0) GO TO 312
     DO 310 I=1,IN
     TX=(U(LM1+I)-U(LP1+I))*CNM(KO+I)- V(LP1+I) *SNM(KO+I)+TX
     TY=(U(LM1+I)+U(LP1+I))*SNM(KO+I)- V(LP1+I) *CNM(KO+I)+TY
  310 TZ = -U(L+I)*CNM(KO+I) - V(L+I)*SNM(KO+I) + TZ
  312 CONTINUE
     IF(NTE.LT.3)GO TO 340
     DO 330 MP1=3,NTE
     KO= KO+IN
```

```
      LM1= L+1
      L= LP1 +1
      LP1=ICT(MP1) + KONE
      IN= LCT(MP1)
      IF(IN.LE.0) GO TO 330
      DO 320 I=1,IN
      TX=(U(LM1+I)-U(LP1+I))*CNM(KO+I)+(V(LM1+I)-V(LP1+I))*SNM(KO+I)+TX
      TY=(U(LM1+I)+U(LP1+I))*SNM(KO+I)-(V(LM1+I)+V(LP1+I))*CNM(KO+I)+TY
  320 TZ= -U(L+I)*CNM(KO+I) - V(L+I)*SNM(KO+I) + TZ
  330 CONTINUE
  340 IN=LCT(1)
      L= IN + 1
      LP1= L+ 1  + KONE
      L2= ICT(1) + LCT(2) + KONE  + 2
      TXO=0.DO
      TYO=0.DO
      DO 350 I=1,IN
      TXO= -CNM(L-I)*U(L2-I)        + TXO
      TYO=-CNM(L-I)*V(L2-I)     + TYO
  350 TZ= -CNM(L-I)*U(LP1-I) + TZ
      XTL(K) = TXO+TXO+TX
      YTL(K) = TYO+TYO+TY
      TX= DRX
      TY=DRY
      KP1 = K+1
      DO 360 I=1,K
      TX=CGB(I)*XTL(KP1-I)  - SGB(I)*YTL(KP1-I)  + TX
  360 TY=CGB(I)*YTL(KP1-I)  + SGB(I)*XTL(KP1-I)  + TY
      XK=K*KP1
      XPO(ITR)= TX/XK
      YPO(ITR)= TY/XK
      ZPO(ITR)= (TZ+TZ+DRZ)/XK
      IF(NBD.GT.2) CALL ENEXPS(ITR)
      IF(ITR.GE.KTR)  RETURN
      KP1= ITR
      K= ITR-1
      KM1=K-1
      OMGK=KM1
      OMGK=OMG/OMGK
      L=K/2
      ITR=ITR+1
      R2K=XPO(1)*XPO(K)+YPO(1)*YPO(K)+ZPO(1)*ZPO(K)
      RM1K=0.DO
      IF(L.LT.2) GO TO 380
```

```
      DO 370 N=2,L
      R2K=XPO(N)*XPO(KP1-N)+YPO(N)*YPO(KP1-N)+ZPO(N)*ZPO(KP1-N)+R2K
  370 RM1K=RMO(N)*RMO(KP1-N) + RM1K
  380 R2K = R2K+R2K
      RM1K= RM1K+RM1K
      IF((L+L).GE.K) GO TO 390
      R2K= XPO(L+1)**2 + YPO(L+1)**2 + ZPO(L+1)**2 + R2K
      RM1K= RMO(L+1)**2 + RM1K
  390 RPT(K)= R2K
      CGB(K)= -OMGK*SGB(KM1)
      SGB(K)=  OMGK*CGB(KM1)
      RCLA=XPO(1)*CGB(K) + YPO(1)*SGB(K) + XPO(K)*CGB(1) + YPO(K)*SGB(1)
      RSLA=YPO(1)*CGB(K) - XPO(1)*SGB(K) + YPO(K)*CGB(1) - XPO(K)*SGB(1)
      RHOCL= 0.DO
      RHOSL= 0.DO
      RHODL= 0.DO
      RM2K= R2K*RMT(1)
      IF(KM1.LT.2) GO TO 410
      DO 400 N= 2,KM1
      RHOCL= CLB(N)*RMT(KP1-N)  + RHOCL
      RHOSL= SLB(N)*RMT(KP1-N)  + RHOSL
      RHODL= ZPO(N)*RMT(KP1-N)  + RHODL
      RM2K = RPT(N)*RMT(KP1-N)  + RM2K
      RCLA = XPO(N)*CGB(KP1-N) + YPO(N)*SGB(KP1-N) + RCLA
  400 RSLA = YPO(N)*CGB(KP1-N) - XPO(N)*SGB(KP1-N) + RSLA
  410 CLB(K)= RCLA
      SLB(K) = RSLA
      RMT(K) = - RM2K*RMT(1)
      CLT(K)= (CLB(1)*RMT(K)+RCLA*RMT(1)+ RHOCL) * ERD
      SLT(K)= (SLB(1)*RMT(K)+RSLA*RMT(1)+ RHOSL) * ERD
      RZR(K) = (ZPO(1)*RMT(K) + ZPO(K)*RMT(1) + RHODL) * ERD
      RMR(K)= RMT(K) * ERD*ERD
      KO=KJ
      KJ=KO+1
      RMO(K) =  (RMT(K)-RM1K)/RMOTWO
      U(KJ)= RMO(K)
  490 CONTINUE
      XXXX = F(1)
       UA = 0.DO
      UB=0.DO
      VB=0.DO
       IF (I . GE. 0 ) G O TO 1000
       UAM = 0.DO
       DO 500 N = 1, K
```

```
      UA = RZR (N) * U (KJ) + UA
     UAM= RMR(N)*U(KJ) + UAM
     UB= CLT(N)*U(KJ) + UB
     VB= SLT(N)*U(KJ) + VB
  500 KJ = KJ - ONE
      KJ = KO +2
      U(KJ) = UA
      UA = UAM * F( 2)
      IF ( J2 . LE. 2) GO TO 530
      DO 520 J = 3, J2
      UAM = 0
      DO 510 N = 1, K
      UA = RZR (N) * U (KJ) + UA
     UAM= RMR(N)*U(KJ) + UAM
  510 KJ = KJ - ONE
      KJ = KO + J
      U (KJ) = UA
  520 UA = UAM * F(J)
  530 CONTINUE
      DO 540 N = 1,K
      UA = RZR (N) * U (KJ) + UA
  540 KJ = KJ - ONE
  542 J = J2 + 1
      KJ = KO + J
      U(KJ) = UA
  550 IF (J . GE. ONE ) GO TO 570
  551 J = J +1
      KJ = KO + J
      U ( KJ) = UB
      V(KJ) = VB
      XXXX = F(J)
      UA = 0.DO
      UB = 0.DO
      VA = 0.DO
      VB = 0.DO
      GO TO ( 552, 1020, 104C, 1060, 560, 570), I
  552 UAM = 0.DO
      VAM = 0.DO
      DO 554 N = 1, K
      UA = RZR (N) * U (KJ) + UA
      VA = RZR (N) * V(KJ) + VA
     UAM= RMR(N)*U(KJ) + UAM
     VAM= RMR(N)*V(KJ) + VAM
      UB = CLT (N) * U(KJ) - SLT (N) * V (KJ)  + UB
```

```
      VB = CLT (N) * V (KJ) + SLT (N) * U (KJ)  + VB
  554 KJ = KJ - ONE
  555 J = J + 1
      KJ = KO + J
      U(KJ) = UA
      V(KJ) = VA
      UA = UAM * F( J )
      VA = VAM * F(J)
      IF (J . GE. J2) GO TO 560
      J1 = J +1
      DO 558 J = J1,J2
      UAM = 0.D0
      VAM = 0.D0
      DO 556 N = 1,K
      UA = RZR (N) * U (KJ) + UA
      VA = RZR (N) * V(KJ) + VA
     UAM= RMR(N)*U(KJ) + UAM
     VAM= RMR(N)*V(KJ) + VAM
  556 KJ = KJ- ONE
      KJ = KO + J
      U(KJ)= UA
      V(KJ) = VA
      UA = UAM * F(J)
  558 VA = VAM*F(J)
  560 CONTINUE
      DO 562 N= 1,K
      UA = RZR (N) * U (KJ) + UA
      VA = RZR (N) * V(KJ) + VA
  562 KJ = KJ-ONE
  564 J = J2+1
      KJ = KO + J
      U(KJ) = UA
      V(KJ) = VA
      GO TO 550
  570 CONTINUE
     DRX=0.D0
     DRY=0.D0
     DRZ=0.D0
      IF (KDR . LE. 0 ) GO TO 300
C    COMPUTE THE CORRECTION TERMS FOR DRAG
     IF (KA-1)610,610,700
  610 TMA=SQRT(XPO(1)*XPO(1)+YPO(1)*YPO(1))
     TMB=ZPO(1)/TMA
     TMC=ATAN(TMB)
```

```
      XAD= TMC
      TMA=TMA+1.DO
      XRD(KA)=TMA*XPO(KC)-XAD
      YRD(KA)=TMA*YPO(KC)-YAD
      ZRD(KA)=TMA*ZPO(KC)
      TMA=0.DO
      KC=KA
      K=KA/2
      DO 720 I=1,K
      TMA=TMA+XRD(I)*XRD(KC)+YRD(I)*YRD(KC)+ZRD(I)*ZRD(KC)
  720 KC=KC-1
      TMA=TMA+TMA
      IF (KC-K)740,740,730
  730 TMA=TMA+XRD(KC)*XRD(KC)+YRD(KC)*YRD(KC)+ZRD(KC)*ZRD(KC)
  740 VSR(KA)=TMA
      TMB=RPT(KA)
      KC=KB
      IF (KB-1)770,770,750
  750 DO 760 I=2,KB
      TMA=TMA-VRB(I)*VRB(KC)
      TMB=TMB-RPO(I)*RPO(KC)
  760 KC=KC-1
  770 VRB(KA)=TMA/(2.0*VRB(1))
      RPO(KA)=TMB/(2.0*RPO(1))
      IF (KA-2)780,780,790
  780 TMB=ECC*RZR(1)/SQRT(1.0-ECC*RMT(1)*ZPO(1)*ZPO(1))
      TMA=RPO(2)-TMB*(ZPO(1)*RMO(1)*RPO(2)-ZPO(2))
      TMB=CPB*RHO(1)
      HDT=TMA
      GO TO 810
  790 KC=KB-1
      TMA=KB
      TMA=CPB/TMA
      TMB=HDT*RHO(KB)
      TMC=2.DO
      DO 800 I=3,KA
      TMB=TMB+TMC*RPO(I)*RHO(KC)
      TMC=TMC+1.DO
  800 KC=KC-1
  810 RHO(KA)=TMA*TMB
      TMA=0.DO
      KC=KA
      DO 820 I=1,KA
      TMA=TMA+RHO(I)*VRB(KC)
```

```
      TMD=ERD/637816.5D0
      TME=0.D0
      KB=10
      GO TO 630
  620 TMC=ATAN(TMB*(TME+TMF)/(TME+TMF*(1.0-ECC)))
  630 TMF = ERD / DSQRT(1.D0-ECC*SIN(TMC)**2)
      HIT=TMA/COS(TMC)-TMF
      TMC=HIT-TME
      TME=HIT
      IF (TMC)640,670,650
  640 TMC=-TMC
  650 KB=KB-1
      IF (TMC-TMD)670,670,660
  660 IF (KB) 670,670,620
  670 CONTINUE
      DO 680 I=1,NHT
      K=I
      IF (HIT-ALT(I))680,690,690
  680 CONTINUE
  690 CPA=CNA(K)
      CPB=CNB(K)
      CTW=CPB
      XAD=-OMG*YPO(1)
      YAD= OMG*XPO(1)
      XRD(1)=XPO(2)-XAD
      YRD(1)=YPO(2)-YAD
      ZRD(1)=ZPO(2)
      VSR(1)=XRD(1)*XRD(1)+YRD(1)*YRD(1)+ZRD(1)*ZRD(1)
      VRB(1)=DSQRT(VSR(1))
      RPO(1)=DSQRT(RPT(1))
      RHO(1)=CPA*EXP(CPB*HIT)
      RVR(1)=RHO(1)*VRB(1)
      TMA=CDC*RVR(1)
      XDD(1)=XRD(1)*TMA
      YDD(1)=YRD(1)*TMA
      ZDD(1)=ZRD(1)*TMA
      GO TO 840
  700 IF (ITR-KDR)710,710,300
  710 KB=KA-1
      KC=KA+1
      TMA=KB
      TMB=OMG/TMA
      TMC=-TMB*YAD
      YAD= TMB*XAD
```

```
      XAD= TMC
      TMA=TMA+1.D0
      XRD(KA)=TMA*XPO(KC)-XAD
      YRD(KA)=TMA*YPO(KC)-YAD
      ZRD(KA)=TMA*ZPO(KC)
      TMA=0.D0
      KC=KA
      K=KA/2
      DO 720 I=1,K
      TMA=TMA+XRD(I)*XRD(KC)+YRD(I)*YRD(KC)+ZRD(I)*ZRD(KC)
  720 KC=KC-1
      TMA=TMA+TMA
      IF (KC-K)740,740,730
  730 TMA=TMA+XRD(KC)*XRD(KC)+YRD(KC)*YRD(KC)+ZRD(KC)*ZRD(KC)
  740 VSR(KA)=TMA
      TMB=RPT(KA)
      KC=KB
      IF (KB-1)770,770,750
  750 DO 760 I=2,KB
      TMA=TMA-VRB(I)*VRB(KC)
      TMB=TMB-RPO(I)*RPO(KC)
  760 KC=KC-1
  770 VRB(KA)=TMA/(2.0*VRB(1))
      RPO(KA)=TMB/(2.0*RPO(1))
      IF (KA-2)780,780,790
  780 TMB=ECC*RZR(1)/SQRT(1.0-ECC*RMT(1)*ZPO(1)*ZPO(1))
      TMA=RPO(2)-TMB*(ZPO(1)*RMO(1)*RPO(2)-ZPO(2))
      TMB=CPB*RHO(1)
      HDT=TMA
      GO TO 810
  790 KC=KB-1
      TMA=KB
      TMA=CPB/TMA
      TMB=HDT*RHO(KB)
      TMC=2.D0
      DO 800 I=3,KA
      TMB=TMB+TMC*RPO(I)*RHO(KC)
      TMC=TMC+1.D0
  800 KC=KC-1
  810 RHO(KA)=TMA*TMB
      TMA=0.D0
      KC=KA
      DO 820 I=1,KA
      TMA=TMA+RHO(I)*VRB(KC)
```

```
  820 KC=KC-1
      RVR(KA)=TMA
      KC=KA
      TMA=0.D0
      TMB=0.D0
      TMC=0.D0
      DO 830 I=1,KA
      TMD=RVR(I)
      TMA=TMA+TMD*XRD(KC)
      TMB=TMB+TMD*YRD(KC)
      TMC=TMC+TMD*ZRD(KC)
  830 KC=KC-1
      XDD(KA)=CDC*TMA
      YDD(KA)=CDC*TMB
      ZDD(KA)=CDC*TMC
  840 DRX=XDD(KA)
      DRY=YDD(KA)
      DRZ=ZDD(KA)
      K=KA
      GO TO 300
 1000  CONTINUE
       DO 1010 N = 1, K
       UA = RZR (N) * U (KJ) + UA
      UB= CLT(N)*U(KJ) + UB
      VB= SLT(N)*U(KJ) + VB
 1010  KJ = KJ-ONE
       GO TO 542
 1020  CONTINUE
       DO 1030 N = 1,K
       UA = RZR (N) * U (KJ) + UA
       VA = RZR (N) * V(KJ) + VA
       UB = CLT (N) * U(KJ) - SLT (N) * V (KJ)  + UB
       VB = CLT (N) * V (KJ) + SLT (N) * U (KJ)  + VB
 1030  KJ = KJ-ONE
       GO TO 564
 1040  CONTINUE
       DO 1050 N = 1,K
       UB = CLT (N) * U(KJ) - SLT (N) * V (KJ)  + UB
       VB = CLT (N) * V (KJ) + SLT (N) * U (KJ)  + VB
 1050  KJ = KJ-ONE
       GO TO 551
 1060  CONTINUE
       UAM = 0.D0
       VAM = 0.D0
```

```
      DO 1070 N =1,K
      UA = RZR (N) * U (KJ) + UA
      VA = RZR (N) * V(KJ) + VA
     UAM= RMR(N)*U(KJ) + UAM
     VAM= RMR(N)*V(KJ) + VAM
1070  KJ = KJ-ONE
      GO TO 555
     END
```

0448 CARDS

A-3.3 SUBROUTINE VARIEQ

```
      SUBROUTINE VARIEQ
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 ONE
      DIMENSION TMT(9)
COMMON BLOCK - REDUCED FORM
      COMMON /ACOM/
     1CNM(136),SNM(136),F(171),COO,C20,EXTRA(99),UMT(2394),VMT(2394)
     2,LCT(16),ICT(17),ONE
      COMMON /CCOM/
     1XPO(16),YPO(16),ZPO(16),XBD(160),YBD(160),ZBD(160),CGB(14),SGB(14)
     2,CSQ(14),XVR(14),SSQ(14),YVR(14),SCS(14),ZVR(14)
     3,QAV(14),QBV(14),QCV(14),BXB(14),BYB(14),BZB(14)
      COMMON /DCOM/
     1HNS(12),CAV(12),CBV(12),ALT(12),CNA(12),CNB(12),XRD(14),YRD(14),
     2ZRD(14),XDD(14),YDD(14),ZDD(14),RHO(14),RPO(14),RVR(14),HNV(14),
     3VLN(14),VSR(14)
      COMMON /FCOM/
     1AMT(126),BMT(126),EMT(126),FXXMYY(14),TWOFXY(14),FXZ(14),FYZ(14)
     2,SPARE(70)
      COMMON /HCOM/
     1TON,TEN,EPS,TIN,DEL,SPC,PSA,FSA,TOJ,UPD,TOP,DTP
     2,CUD,CUV,CUT,ERD,XMU,ALF,OMG,ECC,CDC,CTW,XIT(6)
      COMMON /ICOM/
     1NBD,NTE,NHT,NSU,NPR,NPS,NOS,ICN,KTR,KDR,KVE,KDV,KSP,ISP,KIN,MPT
      DATA    I,L,KA,KB,KC,KDG   /6*0/
      DATA  AU,AV,BU,BV,CU,CV,CFD,TMQ,CTH,TMR,TMS,TMT /20*0.D0/
COMMON BLOCK COMPLETED
      EQUIVALENCE (TMT(1),TMA),(TMT(2),TMB),(TMT(3),TMC)
      EQUIVALENCE (TMT(4),TMD),(TMT(5),TME),(TMT(6),TMF)
      EQUIVALENCE (TMT(7),TMG),(TMT(8),TMH),(TMT(9),TMI)
C
C INITIALIZE ALL OF THE EXTERNAL ARRAYS.
      KMAX= KVE-2
      IF(KMAX.LE.0) RETURN
      CSQ(1)= CGB(1)**2 - 0.5D0
      SCS(1)= CGB(1)*SGB(1)
      TWOMEG= OMG+OMG
      K= 1
      IO2= 3
      I12= ICT(1) + 2
      I22= ICT(2) + 1
      KDG= KDV-2
      IF(KMAX.LT.KDG) KDG= KMAX
      IF(KDG.LE.0) GO TO 100
```

```
      CTH= CDC*OMG
      CFO=CDC*CTW
      HNV(1)=1.D0/RPO(1)
      VRN(1)=1.D0/VSR(1)
      L=9*KDG
      CALL CLEAR (BMT,L)
C
  100 CONTINUE
      FXXMYY(K)= C00*UMT(I22) + C20*UMT(I22+2)
      TWOFXY(K)= C00*VMT(I22) + C20*VMT(I22+2)
      FXZ(K)   = C00*UMT(I12) + C20*UMT(I12+2)
      FYZ(K)   = C00*VMT(I12) + C20*VMT(I12+2)
      FZZK     = C00*UMT(I02) + C20*UMT(I02+2)
C
      TMA= -0.5D0*FZZK
      TMB= 0.D0
      TMC= 0.D0
      TMF= 0.D0
      KMI= K
      DO 110 I= 1,K
      TMA= CSQ(I)*FXXMYY(KMI) - SCS(I)*TWOFXY(KMI) + TMA
      TMB= SCS(I)*FXXMYY(KMI) + CSQ(I)*TWOFXY(KMI) + TMB
      TMC= CGB(I)*FXZ(KMI)    - SGB(I)*FYZ(KMI)    + TMC
      TMF= SGB(I)*FXZ(KMI)    + CGB(I)*FYZ(KMI)    + TMF
  110 KMI= KMI-1
C
      L= 9*K
      AMT(L)   = FZZK
      AMT(L-4)= -FZZK - TMA
      AMT(L-8)= TMA
      AMT(L-7)= TMB
      AMT(L-5)= TMB
      AMT(L-6)= TMC
      AMT(L-2)= TMC
      AMT(L-3)= TMF
      AMT(L-1)= TMF
      IF(KDG.LE.0) GO TO 500
      CALL CLEAR(TMT,9)
      KA= K-1
  310 KDG=KDG-1
      KB=KA+1
      IF (KA)350,350,320
  320 KC=KA
      DO 330 I=2,KB
```

```
      TMA=TMA-HNV(KC)*RPO(I)
      TMB=TMB-VRN(KC)*VSR(I)
  330 KC=KC-1
      HNV(KB)=TMA*HNV(1)
      VRN(KB)=TMB*VRN(1)
      TMA=0.0D0
      TMB=0.0D0
  350 KC=KB
      DO 360 I=1,KB
      TMA=RVR(KC)
      TMB=TMB+TMA*XPO(I)
      TMC=TMC+TMA*YPO(I)
      TMD=TMD+TMA*ZPO(I)
      TMA=VRN(I)
      TME=TME+TMA*XDD(KC)
      TMF=TMF+TMA*YDD(KC)
      TMG=TMG+TMA*ZDD(KC)
  360 KC=KC-1
      BXB(KB)=TMB
      BYB(KB)=TMC
      BZB(KB)=TMD
      XVR(KB)=TME
      YVR(KB)=TMF
      ZVR(KB)=TMG
      TMR=-OMG*TMF
      TMS=OMG*TME
      CALL CLEAR (TMT,9)
      KC=KB
      DO 380 I=1,KB
      TMQ=HNV(I)
      TMA=TMA+TMQ*BXB(KC)
      TMB=TMB+TMQ*BYB(KC)
      TMC=TMC+TMQ*BZB(KC)
  380 KC=KC-1
      QAV(KB)=CFO*TMA+TMR
      QBV(KB)=CFO*TMB+TMS
      QCV(KB)=CFO*TMC
      TMA=0.0D0
      TMB=0.0D0
      TMC=0.0D0
      KC=KB
      DO 400 I=1,KB
      AU=XRD(KC)
      BU=YRD(KC)
```

```
      CU=ZRD(KC)
      AV=QAV(I)
      BV=QBV(I)
      CV=QCV(I)
      TMA=TMA+AU*AV
      TMB=TMB+BU*AV
      TMC=TMC+CU*AV
      TMD=TMD+AU*BV
      TME=TME+BU*BV
      TMF=TMF+CU*BV
      TMG=TMG+AU*CV
      TMH=TMH+BU*CV
      TMI=TMI+CU*CV
  400 KC=KC-1
      TMQ=CTH*RVR(KB)
      TMB=TMB-TMQ
      TMD=TMD+TMQ
      L=9*KA+1
      DO 410 I=1,9
      AMT(L)= TMT(I) + AMT(L)
  410 L=L+1
      CALL CLEAR (TMT,9)
      IF (KDG)500,500,420
  420 CONTINUE
      TMA=CDC*RVR(KB)
      TME=TMA
      TMI=4.0D0*TMA
      KC=KB
      DU 430 I=1,KB
      AU=YVR(KC)
      BU=ZVR(KC)
      AV=XRD(I)
      BV=YRD(I)
      TMA=TMA+XVR(KC)*AV
      TMB=TMB+AU*AV
      TMC=TMC+BU*AV
      TME=TME+AU*BV
      TMF=TMF+BU*BV
  430 KC=KC-1
      TMD=TMB
      TMG=TMC
      TMH=TMF
      TMI=TMI-TMA-TME
      L=9*KA+1
```

```
      DO 450 I=1,9                                                      106350
      BMT(L)=TMT(I)                                                     106360
  450 L=L+1                                                             106370
  500 CONTINUE                                                          106380
      IF(K.GE.KMAX) RETURN                                              106390
      KM1= K                                                            106400
      TWOMGK= KM1                                                       106410
      TWOMGK= TWOMEG/TWOMGK                                             106420
      K= K+1                                                            106430
      I02= I02 + ONE                                                    106440
      I12= I12 + ONE                                                    106450
      I22= I22 + ONE                                                    106460
      CSQ(K)= -SCS(KM1)*TWOMGK                                          106470
      SCS(K)= CSQ(KM1)*TWOMGK                                           106480
      GO TO 100                                                         106490
      END                                                               106500
                                                                   0192 CARDS
```

A-3.4 SUBROUTINE FINALP

```
      SUBROUTINE FINALP
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C     PURPOSE   TO PRINT FINAL RESULTS AND RESIDUALS
C
C
C     INPUT FILES   IARCIN - ARC STABLE PARAMETERS
C                   IPASIN - PASS STABLE PARAMETERS
C
C     OUTPUT FILES  NONE
C
C
C
C
COMMON BLOCKS
C
C
      COMMON /GENCOM/
     1      CONVG,         DIVRG,         SUMO,          SUM1,
     2      COST,          MXITER,        MXDIVG,        ITERNO,
     3      ITERSW,        IDIVNO,        IUNITS,        IGEOFL,
     4      IAUTOR,        IOINP,         ISIGNO,        IUPDNO,
     5      IPRRES,        NARCS,         NSTA,          MXMEAS,
     6      MXARCS,        MXPARM,        MXSTA,         MXLIN,
     7      IPGNO,         PGID(10)
C
C
      COMMON /TYLE/
     1      TITLE(100)
C
C
      COMMON /TSPARM/
     1      NOTS,          NOTS2,         TSLABL(200),   TSINT(200),
     2      TSCUR(200),    TSIWT(200),    TSCWT(200),    TSDELT(200),
     3      ITSEVL(200),   TS2LAB(200),   TS2INT(200)
C
C
      COMMON /ASPARM/
     1      NOAS,          NOAS2,         ASLABL(200),   ASINT(200),
     2      ASCUR(200),    ASIWT(200),    ASCWT(200),    ASDELT(200),
     3      IASEVL(200),   AS2LAB(200),   AS2INT(200)
C
```

```
C
      COMMON /PSPARM/
     1    NCPS,          NOPS2,         PSLABL(200),   PSINT(200),
     2    PSCUR(200),    PSIWT(200),    PSCWT(200),    PSDELT(200),
     3    IPSEVL(200),   PS2EVL(200),   PS2INT(200)
C
C
      COMMON /CWORK/
     1    IDATA(500),    CATA(500),     ICTL,          NINT,
     2    NFLT
C
C
      COMMON /IONUMB/
     1    MFILE,         ISFILE,        IARCIN,        IPASIN,
     2    IARCOT,        IPASCT,        ICARD,         IRESID,
     3    ITAPE,         IALPHA,        LUA,           LUB,
     4   LUC
C
C
      COMMON /STINFO/
     1    STLABL(100),   STLAT(100),    STLON(100),    STHT(100),
     2    ISTSUR(100),   ISTSID(100)
C
C
      COMMON /ACINFO/
     1    ACLABL(100),   EPOCH(100),    EPOCHD(100),   GHA(100),
     2    IROTAT(100),   IASTAB(100,6), IAPNO(100,6)
C
C
      COMMON /EARTH/ ROTAT, GRHA, RADIUS, GRAV, ECC
C
C
      DIMENSION XI(6), XC(6), XL(6)
      DIMENSION NFLAG(3)
      DATA N601/601/,NFLAG/1,2,3/,DATA2/100.D0/,IX/0/,DATA3/1.D20/
C
C
C  FORMAT STATEMENTS
C
C
 5001 FORMAT('0',39X,' F I N A L   R E S U L T S')
 5002 FORMAT('0', 9X,'COMMENTS  ',9A8,A3,/,(20X,9A8,A3))
 5003 FORMAT(//,10X,'NUMBER OF ITERATIONS PERFORMED',I5)
 5004 FORMAT('0', 9X,'ANALYSIS TERMINATED BY')
```

```
 5005 FORMAT('+',32X,'MAXIMUM ITERATIONS')
 5006 FORMAT('+',32X,'CONVERGENCE')
 5007 FORMAT('+',32X,'DIVERGENCE')
 5008 FORMAT('+',32X,'MAXIMUM DIVERGENT ITERATIONS')
 5009 FORMAT('0', 9X,'CONVERGENGE RATIO  (COST FUNCTION)',F20.8 )
 5010 FORMAT('0', 9X,'PARAMETER   PARAMETER',7X,'APRIORI',11X,
     1    'CALCULATED',11X,'TOTAL',13X,'APRIORI',11X,'CALCULATED',/,
     2    12X,'NUMBER',6X,'LABEL',10X,'VALUE',15X,'VALUE',11X,
     3    'CORRECTION',11X,'SIGMA',15X,'SIGMA')
 5011 FORMAT('0',4X,60('* ') )
 5012 FORMAT('0',35X,'TOTALLY  STABLE  PARAMETERS')
 5013 FORMAT('0',25X,'ARC  STABLE  PARAMETERS   ARC NO.',I4,4X,A8 )
 5014 FORMAT('0',10X,'PASS  STABLE  PARAMETERS   ARC NO.',I4,4X,
     1    A8,6X,'STATION NO.',I4,4X,A8,6X,'PASS NO.',I4)
 5015 FORMAT('0',12X,I3,7X,A8,1P,D18.8,4D19.8 )
 5016 FORMAT('0',25X,'STATE  VECTOR      ARC NO.',I4,4X,A8)
 5017 FORMAT('0', 9X,'EARTH  CENTERED  INERTIAL  COORDINATES')
 5018 FORMAT(////,10X,'EARTH  CENTERED  FIXED  COORDINATES')
 5019 FORMAT('0',20X,'LABEL',12X,'APRIORI',15X,'CALCULATED',14X,
     1    'TOTAL',/,39X,'VALUE',18X,'VALUE',15X,'CORRECTION')
 5020 FORMAT('0',19X,A8,1P,3(7X,D15.8))
C
C
C  DISPLAY COMMENTS AND COST FUNCTION, ETC
C
C
      CALL PAGE
      WRITE (6,5001)
      WRITE (6,5002) TITLE
      WRITE (6,5003) ITERNO
      WRITE (6,5004)
      GO TO ( 11, 12, 13, 14 ), ITERSW
      GO TO 15
   11 CONTINUE
      WRITE (6,5006)
      GO TO 15
   12 CONTINUE
      WRITE (6,5005)
      GO TO 15
   13 CONTINUE
      WRITE (6,5007)
      GO TO 15
   14 CONTINUE
      WRITE (6,5008)
```

```
   15 CONTINUE
      WRITE (6,5009) COST
C
C
C  DISPLAY TOTALLY STABLE PARAMETER RESULTS
C
C
      IF( NOTS .LE. 0 ) GC TO 30
      ICNT = MXLIN + 1
      NEMT=10
      DO 25 I=1,NOTS
      IF( ICNT .LT. MXLIN ) GC TO 22
      CALL PAGE
      WRITE (6,5001)
      WRITE (6,5011)
      WRITE (6,5012)
      WRITE(6,5011)
      WRITE (6,5010)
      ICNT = 11
   22 CONTINUE
      CALL SIGWT (2,SIGI,TSIWT(I) )
      CALL SIGWT(2,SIGC,TSCWT(I)   )
      CORR = TSCUR(I) - TSINT(I)
      WRITE (6,5015) I,TSLABL(I),TSINT(I),TSCUR(I),CORR,SIGI,SIGC
C     WRITE VALUES ON DISK TO UPDATE 601 CARDS
      WRITE (10) N601,TSLABL(I),TSCUR(I)
      ICNT = ICNT + 2
   25 CONTINUE
C
C
C  DISPLAY ARC STABLE PARAMETER RESULTS
C
C
   30 CONTINUE
      IRD = 0
C
C  READ ARC STABLE RECORD
C
   32 CONTINUE
      READ (IARCIN,END=90C) ICTL,NINT,NELT,(IDATA(I),I=1,NINT),
     1     (DATA(I),I=1,NFLT)
      IF( ICTL .NE. 1 ) GC TO 32
      IARC = IDATA(1)
      NOAS = IDATA(2)
```

```
C
C
      IF( NOAS .LE. 0 ) GC TO 60
C
C  STORE ARC STABLE PARAMETERS
C
      DO 34 I=1,NOAS
      IASEVL(I) = IDATA(I+2)
      J = 6*I - 5
      ASLABL(I) = DATA(J)
      ASINT(I) = DATA(J+1)
      ASCUR(I) = DATA(J+2)
      ASIWT(I) = DATA(J+3)
      ASCWT(I) = DATA(J+4)
      ASDELT(I) = DATA(J+5)
   34 CONTINUE
C
C  ISOLATE AND DISPLAY STATE VECTOR
C
      DO 350 I=1,6
      J = IASTAB(IARC,I)
      K = IAPNO(IARC,I)
      GO TC (351,352),J
      GO TC 35
  351 CONTINUE
      XI(I) = TSINT(K)
      XC(I) = TSCUR(K)
      XL(I) = TSLABL(K)
      GO TO 350
  352 CONTINUE
      XI(I) = ASINT(K)
      XC(I) = ASCUR(K)
      XL(I) = ASLABL(K)
  350 CONTINUE
      CALL PAGE
      WRITE (6,5001)
      WRITE (6,5011)
      WRITE (6,5016) IARC,ACLABL(IARC)
      WRITE (6,5011)
      WRITE (6,5017)
      WRITE (6,5019)
      DO 354 I=1,6
      CORR = XC(I) - XI(I)
      WRITE (6,5020) XL(I),XI(I),XC(I),CORR
```

```
  354 CONTINUE
      IF( IROTAT(IARC) .LE. 0 ) GO TO 35
      WRITE (6,5018)
      WRITE (6,5019)
      GRHA = GHA(IARC)
      CALL ROTFIX(XI,EPOCH(IARC))
      CALL ROTFIX(XC,EPOCH(IARC))
      DO 356 I=1,6
      CORR = XC(I) - XI(I)
      WRITE (6,5020) XL(I), XI(I),XC(I), CORR
  356 CONTINUE
C
C
   35 CONTINUE
      ICNT = MXLIN + 1
      DO 38 I=1,NOAS
      IF( ICNT .LT. MXLIN ) GC TO 36
      CALL PAGE
      WRITE (6,5001)
      WRITE (6,5011)
      WRITE (6,5013) IARC,ACLABL(IARC)
      WRITE (6,5011)
      WRITE (6,5010)
      ICNT = 11
   36 CONTINUE
      CALL SIGWT(2,SIGI,ASIWT(I))
      CALL SIGWT(2,SIGC,ASCWT(I))
      CORR = ASCUR(I) - ASINT(I)
      WRITE (6,5015) I,ASLABL(I),ASINT(I),ASCUR(I),CORR,SIGI,SIGC
      WRITE (10) N601,ASLABL(I),ASCUR(I)
C     WRITE ON DISK TO UPCATE 601 CARDS
      ICNT = ICNT + 2
   38 CONTINUE
C
C
C  DISPLAY PASS STABLE PARAMETER RESULTS
C
C
   60 CONTINUE
      IF( IRD .NE. 0 ) GO TC 66
   62 CONTINUE
      REAC (IPASIN,END=900) ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),
     1      (DATA(I),I=1,NFLT)
      IF( ICTL .NE. 1 ) GC TO 62
```

```
      KARC = IDATA(1)
      ISTA = IDATA(2)
      IPASS = IDATA(3)
      NOPS = IDATA(4)
C
C  STORE PASS STABLE PARAMETERS
      IF( NOPS .LE. 0 ) GO TO 66
      DO 64 I=1,NOPS
      IPSEVL(I) = IDATA(I+4)
      J = 6*I - 5
      PSLABL(I) = DATA(J)
      PSINT(I) = DATA(J+1)
      PSCUR(I) = DATA(J+2)
      PSIWT(I) = DATA(J+3)
      PSCWT(I) = DATA(J+4)
      PSDELT(I) = DATA(J+5)
   64 CONTINUE
C
C
   66 CONTINUE
      IF( KARC .EQ. IARC ) GO TO 70
      IRD = 1
      GO TO 32
C
C
   70 CONTINUE
      IF( NOPS .LE. 0 ) GO TO 62
      ICNT = MXLIN + 1
      NEMT=10
      DO 76 I=1,NOPS
      IF( ICNT .LE. MXLIN ) GO TO 74
      CALL PAGE
      WRITE (6,5001)
      WRITE (6,5011)
      WRITE (6,5014 ) IARC,ACLABL(IARC),ISTA,STLABL(ISTA),IPASS
      WRITE (6,5011)
      WRITE (6,5010)
      ICNT = 11
   74 CONTINUE
      CALL SIGWT(2,SIGI,PSIWT(I))
      CALL SIGWT( 2,SIGC,PSCWT(I))
      CORR = PSCUR(I) - PSINT(I)
      WRITE (6,5015) I,PSLABL(I),PSINT(I),PSCUR(I),CORR,SIGI,SIGC
      MN=ISTA*2-2+I
```

```
      WRITE (10) N601,PSLABL(I),PSCUR(I)
C      WRITE CN DISK TO UPDATE 601 CARDS
      ICNT = ICNT + 2
   76 CONTINUE
      GO TO 62
C
C
C  END OF PRINT OUT
C
C
  900 CONTINUE
      REWIND IARCIN
      REWIND IPASIN
C
C
      CALL RESID
C
C
      RETURN
      END
```

0331 CARDS

A-3.5 SUBROUTINE RESID

A-3.5.1 STANDARD SUBROUTINE RESID

```
      SUBROUTINE RESID
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL*1 LFILE,GRAPH,SYM,SI,BLNK,SO
C
C
C     PURPOSE   CCMPUTE, CISPLAY, AND OUTPUT MEASUREMENT RESIDUALS
C
C
C
C     COMMON BLOCKS
      COMMON /COMSOL/    U(10C,100),   C(100),       AUTOR(20),
     1    ASUM(20),      APROC(20),    ANUMB(20),    CMW(20),
     2    EPS(20),       EPSL(20),     B(20,40),     BL(20,40),
     3    RES(20),       E,            B1,           B2,
     4    Z,             TIME,         DAYS,         WT,
     5    ALP,           DIS,          SIG,          TEST,
     6    YY,
     A    NBP(20,40),    NBPL(20,40),  NB(20),       NBL(20),
     B    NMSNO(20),     NMEAS,        NSIZEU,       NSIZEB,
     C    NSIZEM,        NMSPT,        NPAR,         IARC,
     D    ISTA,          IPASS,        JARC,         JSTA,
     E    JPASS,         KARC,         KSTA,         KPASS,
     F    IAFLAG,        IPFLAG,       NPARS,        MNO,
     G    IPNO,          ITY,          IA,           IB,
     H    LA,            LB,           IPRU,         IAR,
     I    ICNT,          IWORK(20)
      COMMCN /TSPARM/    NOTS,         NOTS2,        TSLABL(200),
     1    TSINT(200),    TSCUR(200),   TSIWT(200),   TSCWT(200),
     2    TSDELT(200),   ITSEVL(200),  TS2LAB(200),  TS2INT(200)
C
      COMMCN /ASPARM/    NOAS,         NOAS2,        ASLABL(200),
     1    ASINT(200),    ASCUR(200),   ASIWT(200),   ASCWT(200),
     2    ASDELT(200),   IASEVL(200),  AS2LAB(200),  AS2IWT(200)
C
      COMMON /PSPARM/    NOPS,         NOPS2,        PSLABL(200),
     1    PSINT(200),    PSCUR(200),   PSIWT(200),   PSCWT(200),
     2    PSDELT(200),   IPSEVL(200),  PS2LAB(200),  PS2IWT(200)
C
      COMMON /CWORK/     IDATA(500),   DATA(500),    ICTL,
     1    NINT,          NFLT
C
      COMMCN /MSINFO/    XMLABL(300),  MSTYPE(300)
C
```

```
      COMMON /ACINFO/   ACLABL(100),   EPOCH(100),     EPOCHD(100),
     1   GHA(100),      IROTAT(100),   IASTAB(100,6),  IAPNO(100,6)
C
      COMMON /STINFO/   STLABL(100),   STLAT(100),     STLON(100),
     1   STHT(100),     ISTSUR(100),   ISTSID(100)
C
      COMMON /GENCOM/
     1CONVG       ,DIVRG       ,SUMO        ,SUM1        ,
     2COST        ,MXITER      ,MXDIVG      ,ITERNO      ,
     AITERSW      ,
     3IDIVNO      ,IUNITS      ,IGEOFL      ,IAUTOR      ,
     4IDINP       ,ISIGNO      ,IUPDNO      ,IPRRES      ,
     5NARCS       ,NSTA        ,MXMEAS      ,MXARCS      ,
     6MXPARM      ,MXSTA       ,MXLIN       ,IPGNO       ,
     7             PGIC(10)
C
      COMMON /CDEBUG/   IDEFLG(2),     IPRFLG(2),      ISLFLG(2)
C
      COMMON /TYLE/     TITLE(100)
C
      COMMON /IONUMB/   MFILE,         ISFILE,         IARCIN,
     1   IPASIN,        IARCCT,        IPASOT,         ICARD,
     2   IRESID,        ITAPE,         IALPHA
C
C
      DIMENSION XMSNA(20)
      DIMENSION GRAPH(83),K1(2),SYM(2)
      DATA SYM/'L','M'/,GRAPH/83*' '/
      DATA SI/'I'/,BLNK/' '/,SO/'*'/
      DATA ISEQ/0/
      DATA XMSNA/'RANGE   ','AZ COS E','ELEV    ','RA COS D',
     1           'DECLIN  ','COS ALPH','COS BETA','X 30    ',
     2           'Y 30    ','ALTIMETR','R DOT   ','AZ DOT  ',
     3           'ELEV DCT','X 85    ','Y 85    ','        ',
     4           '        ','        ','        ','        '/
C
C     FORMAT STATEMENTS
C
 5001 FORMAT('0',20X,'M E A S U R E M E N T   R E S I D U A L S',
     1   19X,'ITERATION NO.',I4)
 5002 FORMAT('0',10X,'DESIGNATION    MEASUREMENT   MEASUREMENT',4X,
     1      'MEASUREMENT',13X,'ALPHA',/,29X,'LABEL',8X,'NUMBER',10X,
     2      'TYPE',18X,'USED')
 5003 FORMAT('0',10X,'DESIGNATION    MEASUREMENT   MEASUREMENT',4X,
```

```
     1      'MEASUREMENT',/,29X,'LABEL',8X,'NUMBER',10X,'TYPE')
 5004 FORMAT('0',12X,'MEAS',I2,8X,A8,8X,I4,7X,I3,' - ',A8,5X,D15.8)
 5005 FORMAT('0','YR MO DA HR MI  SEC',6X,'MEAS',I2,9X,'MEAS',I2)
 5006 FORMAT('0',4X,'NO. OF POINTS',4X,F11.0,5(7X,F11.0),/,
     1      (15X,6F18.0) )
 5007 FORMAT('0',8X,'MEAN',6X,6(3X,D15.8),/,(19X,6(3X,D15.8)))
 5008 FORMAT('0','STANDARD DEVIATION',6(3X,D15.8),/,(19X,6(3X,D15.8)))
 5009 FORMAT('0',22X,'ARC NO.',I4,3X,A8,3X,'STATION NO.',I4,3X,A8,
     1      3X,'PASS NO.',I4)
 5010 FORMAT(' ',I2,4I3,F6.2)
 5011 FORMAT ('+',21X,D12.4)
 5012 FORMAT ('+',35X,D12.4)
 5013 FORMAT('+',57X,D15.8)
 5014 FORMAT('+',75X,D15.8)
 5015 FORMAT('+',93X,D15.8)
 5016 FORMAT('+',111X,D15.8)
 5017 FORMAT(24X)
 5801 FORMAT('1',10X,'PREMATURE END OF FILE ENCOUNTERED ON IARCIN,',
     1      'PROGRAM TERMINATED')
 5802 FORMAT('1',10X,'PREMATURE END OF FILE ENCOUNTERED ON IPASIN,',
     1      'PROGRAM TERMINATED')
C
C
C
C     INITIALIZATION
      REWIND ISFILE
      REWIND IARCIN
      REWIND IPASIN
      REWIND IALPHA
C        REWIND IRESID
      REWIND IARCOT
C
C     CLEAR ALL STORAGE AREAS
      DO   5 I=1,NSIZEM
      NB(I) = 0
      NBL(I) = 0
      CMW(I) = 0.D0
      EPS(I) = 0.D0
      EPSL(I) = 0.D0
      AUTOR(I) = 0.D0
      ASUM(I) = 0.D0
      APROD(I) = 0.D0
      ANUMB(I) = 0.D0
      NMSNC(I) = 0
```

```
      DO   5 J=1,NSIZEB
      NBP(I,J) = 0.D0
      NBPL(I,J)= 0.D0
      B(I,J) = 0.D0
      BL(I,J) = 0.D0
    5 CONTINUE
      COVAR=0.D0
C
C
      LFILS=.FALSE.
      GRAPH(42)=SI
C     SET FLAGS FOR 1ST ARC + PASS
      IASOLN = 0
      IPSOLN = 0
C
C     READ CONTROL FILE
C
  100 CONTINUE
      READ (ISFILE,END=4000)ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),
     1   (DATA(I),I=1,NFLT)
C
      GO TO ( 1000, 2000, 3000 ), ICTL
      GO TO 100
C
C
C
C     BEGINNING OF AN ARC
C
C
 1000 CONTINUE
C
C     TEST FOR FIRST ARC
      IF(IASOLN .NE. 0 )   GO TO 1002
      IASOLN = 1
      GO TO 1010
C
C     GO TO PRINT STATISTICS ROUTINE .
 1002 CONTINUE
      ISTAT = 1
      GO TO 6000
 1004 CONTINUE
C
C     READ + STORE ARC PARAMETERS
C
```

```
 1010 CONTINUE
      IARC = IDATA(1)
 1012 CONTINUE
      READ (IARCIN,END=4100) ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),
     1     (DATA(I),I=1,NFLT)
C
C     IF NOT TYPE 1 RECORD,READ AGAIN
      IF(ICTL .NE. 1 )  GC TO 1012
C
C     STORE PARAMETERS
      NOAS = IDATA(2)
      KARC = IDATA(1)
C
C     SET FLAGS FOR FIRST PASS OF AN ARC
      IPSOLN = 0
      IF( NOAS .LE. 0)  GC TO 1018
      DO 1016 I=1,NOAS
      IASEVL(I) = IDATA(I+2)
      J = 6*I - 5
      ASLABL(I) = DATA(J)
      ASINT(I)  = DATA(J+1)
      ASCUR(I)  = DATA(J+2)
      ASIWT(I)  = DATA(J+3)
      ASCWT(I)  = DATA(J+4)
      ASDELT(I)  = DATA(J+5)
 1016 CONTINUE
 1018 CONTINUE
      GO TO 100
C
C
C     BEGINNING OF A PASS
C
C
 2000 CONTINUE
C
C     IF NOT FIRST PASS OF AN ARC, PRINT STATISTICS FROM LAST PASS
      IF( IPSOLN .NE. 0 )  GO TO 2002
      IPSOLN = 1
      GO TC 2008
 2002 CONTINUE
      ISTAT =2
      GO TO 6000
 2004 CONTINUE
C
```

```
C     STORE MEASUREMENT  INFO FOR THIS PASS
C
 2008 CONTINUE
      KARC = IDATA(1)
      ISTA = IDATA(2)
      IPASS = IDATA(3)
      NMEAS = IDATA(4)
      DO 2010 I=1,NMEAS
      NMSNO(I) = IDATA(I+4)
      CMW(I) = DATA(I)
 2010 CONTINUE
C
C     IF FINAL PRINT AND AUTOREGRESSIVE CONSTANTS USED,READ IALPHA
C        AND SCORE CONSTANTS
C
      IF ((ITERSW .EQ. 0) .OR.
     1    (IAUTOR .NE. 0)) GO TO 2020
      READ (IALPHA) ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),
     1   (DATA(I),I=1,NFLT)
      DO 2016 I=1,NMEAS
      DO 2014 J=1,NMEAS
      IF( IDATA(I+4) .NE. NMSNO(J)) GO TO 2014
      AUTOR(J) = DATA(I)
      GO TO 2016
 2014 CONTINUE
 2016 CONTINUE
C
C     READ PASS STABLE PARAMETERS
C
 2020 CONTINUE
      READ (IPASIN,END=4101) ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),
     1  (DATA(I),I=1,NFLT)
      IF( ICTL .NE. 1 ) GO TO 2020
C
C     STORE PARAMETER VALUES FROM TYPE 1 RECORD
      JARC = IDATA(1)
      JSTA = IDATA(2)
      JPASS = IDATA(3)
      NOPS = IDATA(4)
      IF(NOPS .LE. 0 )  GO TO 2030
      DO 2024 I=1,NOPS
      IPSEVL(I) = IDATA(I+4)
      J = 6*I - 5
      PSLABL(I) = DATA(J)
```

```
      PSINT(I) = DATA(J+1)
      PSCUR(I) = DATA(J+2)
      PSIWT(I) = DATA(J+3)
      PSCWT(I) = DATA(J+4)
      PSDELT(I) = DATA(J+5)
 2024 CONTINUE
C
C     PRINT HEADINGS AND MEASUREMENT LEGEND
C
 2030 CONTINUE
      CALL PAGE
      WRITE (6,5001) ITERNO
      WRITE (6,5009) JARC,ACLABL(IARC),ISTA,STLABL(ISTA),IPASS
      IF ((ITERSW .NE. 0) .AND.
     1    (IAUTOR .EQ. 0)) GO TO 2036
C
C     NO AUTOREGRESSIVE CONSTANTS PRINTED
      WRITE (6,5003)
      DO 2034 I=1,NMEAS
      WRITE (6,5004) I, XMLABL(NMSNO(I)),NMSNO(I),MSTYPE(NMSNO(I)),
     1    XMSNA(MSTYPE(NMSNO(I)))
 2034 CONTINUE
      GO TO 2040
C
C     PRINT ALPHAS YOO
 2036 CONTINUE
      WRITE (6,5002)
      DO 2038 I=1,NMEAS
      WRITE (6,5004) I,XMLABL(NMSNO(I)),NMSNO(I),MSTYPE(NMSNO(I)),
     1    XMSNA(MSTYPE(NMSNO(I))),AUTOR(I)
 2038 CONTINUE
C
C
 2040 CONTINUE
      WRITE (6,5005) (I,I=1,NMEAS)
      ICNT = 10 + 2 *NMEAS
      GO TO 100
C
C
C     MEASUREMENT POINT RECORC
C
C
 3000 CONTINUE
C
```

```
C     CHECK FOR PAGE OVERFLOW
      IF(ICNT .LT. MXLIN ) GO TO 3002
      CALL PAGE
      WRITE (6,5001) ITERNO
      WRITE (6,5009) IARC,ACLABL(IARC),ISTA,STLABL(ISTA),IPASS
      WRITE (6,5005) (I,I=1,NMEAS)
      ICNT = 7
 3002 CONTINUE
C
C     CONVERT DAYS + TIME TO YR,MON,DAY,HR,MIN,SEC  + DISPLAY
      CALL DAYHMS(DATA(1),DATA(2),IYR,IMO,IDA,IHR,IMIN,SEC)
      WRITE (6,5010) IYR, IMO, IDA, IHR, IMIN, SEC
C
C     PROCESS THE MEASUREMENT POINT RECORD
C
C     CLEAR STORAGE AREAS
      DO 3004 I=1,NSIZEM
      EPS(I) = 0.DO
      RES(I) = 0.DO
      NB(I) = 0
      IWORK(I) = 0
      DO 3004 J=1,NSIZEB
      B(I,J) = 0
 3004 CONTINUE
C
C     STORE MEASUREMENT DISCREPENCIES
      NMSPT = IDATA(1)
      NPAR = IDATA(2)
      DO 3008 I=1,NMSPT
      DO 3006 J=1,NMEAS
      IF(IDATA(I+2) .NE. NMSNC(J) ) GO TO 3006
      IWORK(J) = NMSNC(J)
      EPS(J) = DATA(I+2)
      GO TO 3008
 3006 CONTINUE
C     NO MATCH ERROR
 3008 CONTINUE
C
C     STORE PARTIALS BY MEASUREMENT NO INDICATOR
C
      IF(NPAR .LE. 0 ) GO TO 3016
      DO 3012 I=1,NPAR
C     CALCULATE PARAMETER NO
      IPNC = IDATA(I*3+NMSPT +1)
```

```
      ITY = IDATA(I*3+NMSPT)
      MNO = IDATA(I*3+NMSPT+2)
      IF( ITY .GE. 2 ) IPNO = IPNO + NOTS
      IF( ITY .EQ. 3 ) IPNO = IPNO+NOAS
C     FIND MEAS NO INDICATOR
      DO 3010  J=1,NMEAS
      IF( MNO .NE. NMSNO(J) ) GO TO 3010
      NB(J) = NB(J) + 1
      B(J,NB(J)) = DATA(NMSPT+I+2)
      NBP(J,NB(J)) = IPNO
      GO TO 3012
 3010 CONTINUE
C     NO MATCH ERROR
 3012 CONTINUE
C
C     CALCULATE RESIDUALS AND DISPLAY
C        OUTPUT RESIDUALS IF FINAL CALL TO RESID
C
 3016 CONTINUE
      DO 3036 I=1,NMEAS
C     SKIP TO NEXT LINE IF LINE IS FULL
      J = I/6*6 + 1
      IF((I.NE.1).AND.(I.EQ.J)) WRITE(6,5017)
      IF( EPS(I) .EQ. 0.DC )  GO TO 3036
      E = EPS(I)
C     CHECK FOR USE OF AUTOREGRESSIVE CONSTANTS
      IF((ITERSW .NE.0).AND.(IAUTOR.EQ.0)) E = E - AUTOR(I)*EPSL(I)
      RES(I) = E
C     IF NO PARTIALS, DISPLAY AND GO TO NEXT MEAS
      IF( NB(I) .LE. 0 ) GO TO 3034
      NPARS = NB(I)
      DO 3032 J=1,NPARS
      B1 = B(I,J)
C     CHECK FOR USE OF AUTOREGRESSIVE CONSTANTS
      IF((ITERSW .NE.0).AND.(IAUTOR.EQ.0)) B1 = B1 -AUTOR(I)*BL(I,J)
      IPNO = NBP(I,J)
      IF( IPNO .GT. NOTS ) GO TO 3020
      RES(I) = RES(I) - TSDELT(IPNO) * B1
      GO TO 3032
 3020 CONTINUE
      IPNO = IPNO - NOTS
      IF( IPNO .GT. NOAS ) GO TO 3022
      RES(I) = RES(I) - ASDELT(IPNO) * B1
      GO TO 3032
```

```
 3022 CONTINUE
      IPNC = IPNO - NOAS
      RES(I) = RES(I) - PSDELT(IPNO) * B1
 3032 CONTINUE
C     PRINT CALCULATED RESIDUALS
 3034 CONTINUE
      J = I
 3024 IF( J .LE. 6 ) GO TC 3026
      J = J - 6
      GO TO 3024
 3026 CONTINUE
      GO TO (3041,3042,3043,3044,3045,3046),J
 3041 CONTINUE
      WRITE (6,5011) RES(I)
      GO TO 3050
 3042 CONTINUE
      WRITE (6,5012) RES(I)
      GO TO 3050
 3043 CONTINUE
      WRITF (6,5013) RES(I)
      GO TO 3C50
 3044 CONTINUE
      WRITE (6,5014) RES(I)
      GO TO 3050
 3045 CONTINUE
      WRITE (6,5015) RES(I)
      GO TO 3C50
 3046 CONTINUE
      WRITE (6,5016) RES(I)
 3050 CONTINUE
C
C     ACD TO COUNTS, SUMS, PRODUCTS
      ANUMB(I) = ANUMB(I) + 1.DO
      ASUM(I) = ASUM(I) + RES(I)
      APRCD(I)=APROD(I)+RES(I)**2
 3036 CONTINUE
      COVAR=COVAR +  RES(1)* RES(2)
      DO 3090 I=1,2
      IF (RES(I)) 3070,3070,3C80
 3070 KK=(RES(I)-.05D-04)/(-.2D-04)
      IF (KK .GT. 40) KK=41
      GRAPH (42-KK)=SYM(I)
      K1(I)=42-KK
      GO TO 3090
```

```
 3080 KK=(RES(I)+.05D-04)/.2D-04
      IF (KK .GT. 40) KK=41
      GRAPH(42+KK)= SYM(I)
      K1(I)=42+KK
 3090 CONTINUE
      IF (K1(1) .NE. K1(2)) GC TO 3095
      GRAPH(K1(1))=SO
 3095 WRITE (6,3091) GRAPH
 3091 FORMAT ('+',49X,83A1)
      GRAPH(K1(1))=BLNK
      GRAPH(K1(2))=BLNK
      IF (K1(1) .EQ. 42) GRAPH(42)=SI
      IF (K1(2) .EQ. 42) GRAPH(42)=SI
      ICNT = ICNT + (NMEAS+5)/6
C
C     IF FINIAL CALL TO RESID, CUTPUT RESIDUALS
C
      IF( ITERSW .EQ. 0 )  GO TO 3060
C
      ICTL = 1
      NINT = NMEAS + 4
      NFLT = NMEAS + 2
C      WRITE(IRESID) ICTL,NINT,NFLT,IARC,ISTA,IPASS,NMEAS,
C    *   (IWORK(I),I=1,NMEAS),DATA(1),DATA(2),(RES(I),I=1,NMEAS)
C
C     IF USING AUTOREGRESSIVE CONSTANTS MOVE CURRENT VALUES TO PREVIOUS
 3C60 CONTINUE
      IF((ITERSW.EQ.0).OR.(IAUTOR.NE.0))  GO TO 100
      DO 3C66 I=1,NSIZEM
      NBL(I) = NB(I)
      EPSL(I) = EPS(I)
      DO 3066 J=1,NSIZEB
      BL(I,J) = B(I,J)
      NBPL(I,J) = NBP(I,J)
 3C66 CONTINUE
      GO TO 100
C
C
C     DISPLAY RESIDUAL STATISTICS
C
C
 6000 CONTINUE
      WRITE (6,5006) (ANUMB(I),I=1,NMEAS)
C     COMPUTE MEAN AND STANDARD DEVIATION
```

```
      DO 6004 I=1,NMEAS
      IF( ANUMB(I) .LE. 1.DO )  GO TO 6002
      APROD(I) = DSQRT((APRCD(I)-ASUM(I)**2/ANUMB(I))/(ANUMB(I)-1.DO))
      ASUM(I) = ASUM(I)/ANUMB(I)
      GO TC 6004
 6002 APRCD(I) = 0.DO
 6004 CONTINUE
      WRITE (6,5007) (ASUM(I),I=1,NMEAS)
      WRITE (6,5008) (APRCD(I),I=1,NMEAS)
      IF (ANUMB(1) .LE. 1.DO)  SCV=0.DC
      IF (ANUMB(1) .GT. 1.DO) SCV= (COVAR-ANUMB(1)*ASUM(1)*ASUM(2))/
     *          ((ANUMB(1)-1.DC)*APRCD(1)*APROD(2))
      WRITE (6,8000) SCV
 8000 FORMAT ('0','CORRELATION COEFFICIENT',3X,F6.2)
C     WRITE RESIDUALS ON SUMMARY FILE
 9000 FORMAT (1H1,30X,'M E A S U R E M E N T  R E S I D U A L  S U M M A
     *R Y'//)
 9002 FORMAT (3X,'STATION',6X,'ARC  PASS  YR MO DAY  HR MIN SEC',17X,
     *        'MEAN',22X,'STANDARD DEVIATION',9X,'CORRELATION'/
     *        2X,'NO.  NAME',45X,'MEAS 1',11X,'MEAS 2',11X,'MEAS 1',
     *        11X,'MEAS 2',5X,'COEFFICIENT'//)
      IF (LFILE) GO TO 9050
      LFILE =.TRUE.
      WRITE (34,9000)
      WRITE (34,9002)
 9001 FORMAT (1X,I3,3X,A8,3(2X,I3),2I3,I5,I3,1X,F5.2,1X,4(D15.8,2X),
     *        F7.2)
 9050 WRITE (34,9001) ISTA,STLABL(ISTA),IARC,IPASS,
     *        IYR,IMO,IDA,IHR,IMIN,SEC
     *        ASUM(1),A UM(2),APROD(1),APROD(2) ,SCV
C      WRITE ON DISK TO BE USED BY SUMMARY PRINT PROGRAM
C      CUTPUT :  MEAS 2 LABEL, MEAS 2 NO.,CORRELATION COEFFICIENT,
C                MEAN 1, STD DEV 1, MEAN2, STD DEV 2
      WRITE (10) XMLABL(NMSNO(2)),NMSNO(2),SCV,ANUMB(2),ASUM(1),
     *           APROD(1),ASUM(2),APROD(2)
C
      IF( ITERSW .EQ. 0 ) GO TO 6010 .
      IC = 1
      NI = 4 + NMEAS
      NF = 3 * NMEAS
      WRITE (IARCOT) IC,NI,NF,IARC,ISTA,IPASS,NMEAS,(NMSNO(I),I=1,NMEAS)
     1    ,(ANUMB(I),ASUM(I),APROD(I),I=1,NMEAS)
 6010 CONTINUE
C
```

```
C     CLEAR ARRAYS
      DO 6008 I=1,NSIZEM
      ANUMB(I) = C.DO
      ASUM(I) = 0.DO
      APROD(I) = C.DO
      NMSNO(I) = 0
      NBL(I) = 0
      DO 6008 J= 1,NSIZEB
      BL(I,J) = 0.DO
      NBPL(I,J) =0
 6008 CONTINUE
      COVAR=0.DO
C
      GO TO ( 1004, 2004, 4004 ), ISTAT
C
C
C
C     END OF INPUT DATA
C
 4000 CONTINUE
      ISTAT = 3
      GO TO 6C00
 4004 CONTINUE
      ENDFILE IARCOT
      REWIND IARCOT
C      ENDFILE IRESID
C      REWIND IRESID
      REWIND ISFILE
      REWIND IARCIN
      REWIND IPASIN
      REWIND IALPHA
       ENDFILE 10
      REWIND 10
      RETURN
C
C
C     PREMATURE END OF FILE - IARCIN - ERROR
 4100 CONTINUE
      WRITE (6,5801)
 4103 CONTINUE
      STOP
C
C     PREMATURE END OF FILE - IPASIN - ERROR
 4101 CONTINUE
```

```
      WRITE (6,5802)                                                    136550
      GO TO 4103                                                        136560
      END                                                               136570
```

0576 CARDS

A-3.5.2 SPECIAL SUBROUTINE RESID

```
      SUBROUTINE RESID                                                  130840
      IMPLICIT REAL*8(A-H,O-Z)                                          130850
      LOGICAL*1 LFILE,GRAPH,SYM,SI,BLNK,SO                              130860
C                                                                       130870
C                                                                       130880
C     PURPOSE   COMPUTE, DISPLAY, AND OUTPUT MEASUREMENT RESIDUALS      130890
C                                                                       130900
C                                                                       130910
C                                                                       130920
C     COMMON BLOCKS                                                     130930
      COMMON /COMSOL/    U(110,110),     C(110),         AUTOR(20),
     1    ASUM(20),      APROD(20),      ANUMB(20),      CMW(20),       130950
     2    EPS(20),       EPSL(20),       B(20,40),       BL(20,40),     130960
     3    RES(20),       E,              B1,             B2,            130970
     4    Z,             TIME,           DAYS,           WT,            130980
     5    ALP,           DIS,            SIG,            TEST,          130990
     6    YY,                                                           131000
     A    NBP(20,40),    NBPL(20,40),    NB(20),         NBL(20),       131010
     B    NMSNO(20),     NMEAS,          NSIZEU,         NSIZEB,        131020
     C    NSIZEM,        NMSPT,          NPAR,           IARC,          131030
     D    ISTA,          IPASS,          JARC,           JSTA,          131040
     E    JPASS,         KARC,           KSTA,           KPASS,         131050
     F    IAFLAG,        IPFLAG,         NPARS,          MNO,           131060
     G    IPNO,          ITY,            IA,             IB,            131070
     H    LA,            LB,             IPRU,           IAR,           131080
     I    ICNT,          IWORK(20)                                      131090
      COMMON /TSPARM/    NOTS,           NOTS2,          TSLABL(200),   131100
     1    TSINT(200),    TSCUR(200),     TSIWT(200),     TSCWT(200),    131110
     2    TSDELT(200),   ITSEVL(200),    TS2LAB(200),    TS2INT(200)    131120
C                                                                       131130
      COMMON /ASPARM/    NOAS,           NOAS2,          ASLABL(200),   131140
     1    ASINT(200),    ASCUR(200),     ASIWT(200),     ASCWT(200),    131150
     2    ASDELT(200),   IASEVL(200),    AS2LAB(200),    AS2IWT(200)    131160
C                                                                       131170
      COMMON /PSPARM/    NOPS,           NOPS2,          PSLABL(200),   131180
     1    PSINT(200),    PSCUR(200),     PSIWT(200),     PSCWT(200),    131190
     2    PSDELT(200),   IPSEVL(200),    PS2LAB(200),    PS2IWT(200)    131200
C                                                                       131210
      COMMON /CWORK/     IDATA(500),     DATA(500),      ICTL,          131220
     1    NINT,          NFLT                                           131230
C                                                                       131240
      COMMON /MSINFO/    XMLABL(300),    MSTYPE(300)                    131250
C                                                                       131260
      COMMON /ACINFO/    ACLABL(100),    EPOCH(100),     EPOCHD(100),   131270
```

```
     1    GHA(100),        IROTAT(100),    IASTAB(100,6), IAPNO(100,6)
C
      COMMON /STINFO/     STLABL(100),    STLAT(100),     STLON(100),
     1    STHT(100),      ISTSUR(100),    ISTSID(100)
C
      COMMON /GENCOM/
     1CONVG           ,DIVRG          ,SUMO           ,SUM1           ,
     2COST            ,MXITER         ,MXDIVG         ,ITERNO         ,
     AITERSW          ,
     3IDIVNO          ,IUNITS         ,IGEOFL         ,IAUTOR         ,
     4IOINP           ,ISIGNO         ,IUPDNO         ,IPRRES         ,
     5NARCS           ,NSTA           ,MXMEAS         ,MXARCS         ,
     6MXPARM          ,MXSTA          ,MXLIN          ,IPGNO          ,
     7                 PGID(10)
C
      COMMON /CDEBUG/     IDEFLG(2),      IPRFLG(2),      ISLFLG(2)
C
      COMMON /TYLE/       TITLE(100)
C
      COMMON /IONUMB/     MFILE,          ISFILE,         IARCIN,
     1    IPASIN,         IARCOT,         IPASOT,         ICARD,
     2    IRESID,         ITAPE,          IALPHA
C
C
      DIMENSION ADUM(6)
      DIMENSION XMSNA(20)
      DIMENSION GRAPH(83),K1(2),SYM(2)
      DATA SYM/'L','M'/,GRAPH/83*' '/
      DATA SI/'I'/,BLNK/' '/,SO/'*'/
      DATA ISEQ/0/
      DATA XMSNA/'RANGE   ','AZ COS E','ELEV    ','RA COS D',
     1           'DECLIN  ', 'COS ALPH', 'COS BETA', 'X 30    ',
     2           'Y 30    ', 'ALTIMETR', 'R DOT   ', 'AZ DOT  ',
     3           'ELEV DOT', 'X 85    ', 'Y 85    ', '        ',
     4           '        ', '        ', '        ', '        '/
C
C     FORMAT STATEMENTS
C
 9001 FORMAT ( I4, 1X, A8, I4, 2I3, I4, I3, F7.3, I5, 2D14.5,
     *   2D12.3, I8, I5)
 9002 FORMAT (  79X 'STANDARD' /
     1   52X 'MEAN ERROR' 16X 'DEVIATION' 11X 'NO.  PTS.'/
     2   2X 'STATION' 29X 'PASS' 6X 'MEAS' 10X 'MEAS' 9X 'MEAS'
     3   8X 'MEAS' 7X 'MEAS MEAS' /
```

```
    4   2X 'NO  NAME' 6X 'Y  M  D   H  M  SEC      NO' 8X '1' 13X '2'
    5   12X '1' 11X '2' 11X '1     2'   ///)
 5001 FORMAT('0',10X,'M E A S U R E M E N T   R E S I D U A L S',
    *  10X,'ARC',I3,10X,'ITERATION',I4//)
 5017 FORMAT(24X)
 5801 FORMAT('1',10X,'PREMATURE END OF FILE ENCOUNTERED ON IARCIN,',
    1     'PROGRAM TERMINATED')
 5802 FORMAT('1',10X,'PREMATURE END OF FILE ENCOUNTERED ON IPASIN,',
    1     'PROGRAM TERMINATED')
C
C
C
C     INITIALIZATION
      REWIND ISFILE
      REWIND IARCIN
      REWIND IPASIN
      REWIND IALPHA
C        REWIND IRESID
      REWIND IARCOT
      NUTAPE=35
      NUPASS=36
      NURITE=37
      END FILE NURITE
      IF(ITERSW.EQ.0) GO TO 4
      REWIND 10
    3 READ(10,END=4) N,X,Y
      WRITE(NURITE) N,X,Y
      GO TO 3
    4 REWIND 10
C
C     CLEAR ALL STORAGE AREAS
      DO   5 I=1,NSIZEM
      NB(I) = 0
      NBL(I) = 0
      CMW(I) = 0.D0
      EPS(I) = 0.D0
      EPSL(I) = 0.D0
      AUTOR(I) = 0.D0
      ASUM(I) = 0.D0
      APROD(I) = 0.D0
      ANUMB(I) = 0.D0
      NMSNO(I) = 0
      DO   5 J=1,NSIZEB
      NBP(I,J) = 0.D0
```

```
      NBPL(I,J)= 0.D0
      B(I,J) = 0.D0
      BL(I,J) = 0.D0
    5 CONTINUE
      COVAR=0.D0
C
C
      LFILE=.FALSE.
      GRAPH(42)=SI
C     SET FLAGS FOR 1ST ARC + PASS
      IASOLN = 0
      IPSOLN = 0
C
C     READ CONTROL FILE
C
  100 CONTINUE
      READ (ISFILE,END=4000)ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),
     1   (DATA(I),I=1,NFLT)
C
      GO TO ( 1000, 2000, 3000 ), ICTL
      GO TO 100
C
C
C
C     BEGINNING OF AN ARC
C
C
 1000 CONTINUE
C
      ICNT=MXLIN
      IARC = IDATA(1)
 1012 CONTINUE
      READ (IARCIN,END=4100) ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),
     1    (DATA(I),I=1,NFLT)
C
C     IF NOT TYPE 1 RECORD,READ AGAIN
      IF(ICTL .NE. 1 )  GO TO 1012
C
C     STORE PARAMETERS
      NOAS = IDATA(2)
      KARC = IDATA(1)
C
C     SET FLAGS FOR FIRST PASS OF AN ARC
      IPSOLN = 0
```

```
      IF( NOAS .LE. 0)  GO TO 1018
      DO 1016 I=1,NOAS
      IASEVL(I) = IDATA(I+2)
      J = 6*I - 5
      ASLABL(I) = DATA(J)
      ASINT(I)  = DATA(J+1)
      ASCUR(I)  = DATA(J+2)
      ASIWT(I)  = DATA(J+3)
      ASCWT(I)  = DATA(J+4)
      ASDELT(I)  = DATA(J+5)
 1016 CONTINUE
 1018 CONTINUE
      GO TO 100
C
C
C     BEGINNING OF A PASS
C
C
 2000 CONTINUE
C     STORE MEASUREMENT  INFO FOR THIS PASS
C
 2008 CONTINUE
      KARC = IDATA(1)
      ISTA = IDATA(2)
      IPASS = IDATA(3)
      NMEAS = IDATA(4)
      DO 2010 I=1,NMEAS
      NMSNO(I) = IDATA(I+4)
      CMW(I) = DATA(I)
 2010 CONTINUE
C
C
C     READ PASS STABLE PARAMETERS
C
 2020 CONTINUE
      READ (IPASIN,END=4101) ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),
     1  (DATA(I),I=1,NFLT)
      IF( ICTL .NE. 1 ) GO TO 2020
C
C     STORE PARAMETER VALUES FROM TYPE 1 RECORD
      JARC = IDATA(1)
      JSTA = IDATA(2)
      JPASS = IDATA(3)
      NOPS = IDATA(4)
```

C.6

```
      IF(NOPS .LE. 0 )  GO TO 100
      DO 2024 I=1,NOPS
      IPSEVL(I) = IDATA(I+4)
      J = 6*I - 5
      PSLABL(I) = DATA(J)
      PSINT(I) = DATA(J+1)
      PSCUR(I) = DATA(J+2)
      PSIWT(I) = DATA(J+3)
      PSCWT(I) = DATA(J+4)
      PSDELT(I) = DATA(J+5)
 2024 CONTINUE
C
      GO TO 100
C
C
C     MEASUREMENT POINT RECORD
C
C
 3000 CONTINUE
C
C     CHECK FOR PAGE OVERFLOW
      IF(ICNT .LT. MXLIN ) GO TO 3002
      CALL PAGE
      WRITE(6,5001) IARC,ITERNO
      WRITE(6,9002)
      ICNT = 7
 3002 CONTINUE
C
C     CONVERT DAYS + TIME TO YR,MON,DAY,HR,MIN,SEC  + DISPLAY
      CALL DAYHMS(DATA(1),DATA(2),IYR,IMO,IDA,IHR,IMIN,SEC)
C
C     PROCESS THE MEASUREMENT POINT RECORD
C
C     CLEAR STORAGE AREAS
      DO 3004 I=1,NSIZEM
      EPS(I) = 0.D0
      RES(I) = 0.D0
      NB(I) = 0
      IWORK(I) = 0
      DO 3004 J=1,NSIZEB
      B(I,J) = 0
 3004 CONTINUE
C
C     STORE MEASUREMENT DISCREPENCIES
```

```
      NMSPT = IDATA(1)
      NPAR = IDATA(2)
      DO 3008 I=1,NMSPT
      DO 3006 J=1,NMEAS
      IF(IDATA(I+2) .NE. NMSNO(J) ) GO TO 3006
      IWORK(J) = NMSNO(J)
      EPS(J) = DATA(I+2)
      GO TO 3008
 3006 CONTINUE
C     NO MATCH ERROR
 3008 CONTINUE
C
C     STORE PARTIALS BY MEASUREMENT NO INDICATOR
C
      IF(NPAR .LE. 0 ) GO TO 3016
      DO 3012 I=1,NPAR
C     CALCULATE PARAMETER NO
      IPNO = IDATA(I*3+NMSPT +1)
      ITY = IDATA(I*3+NMSPT)
      MNO = IDATA(I*3+NMSPT+2)
      IF( ITY .GE. 2 ) IPNO = IPNO + NOTS
      IF( ITY .EQ. 3 ) IPNO = IPNO+NOAS
C     FIND MEAS NO INDICATOR
      DO 3010  J=1,NMEAS
      IF( MNO .NE. NMSNO(J) ) GO TO 3010
      NB(J) = NB(J) + 1
      B(J,NB(J)) = DATA(NMSPT+I+2)
      NBP(J,NB(J)) = IPNO
      GO TO 3012
 3010 CONTINUE
C     NO MATCH ERROR
 3012 CONTINUE
C
C     CALCULATE RESIDUALS AND DISPLAY
C        OUTPUT RESIDUALS IF FINAL CALL TO RESID
C
 3016 CONTINUE
      DO 3036 I=1,NMEAS
C     SKIP TO NEXT LINE IF LINE IS FULL
      J = I/6*6 + 1
      IF((I.NE.1).AND.(I.EQ.J)) WRITE(6,5017)
      IF( EPS(I) .EQ. 0.DO )  GO TO 3036
      E = EPS(I)
C     CHECK FOR USE OF AUTOREGRESSIVE CONSTANTS
```

```
      IF((ITERSW .NE.0).AND.(IAUTOR.EQ.0)) E = E - AUTOR(I)*EPSL(I)
      RES(I) = E
C     IF NO PARTIALS, DISPLAY AND GO TO NEXT MEAS
      IF( NB(I) .LE. 0 ) GO TO 3034
      NPARS = NB(I)
      DO 3032 J=1,NPARS
      B1 = B(I,J)
C     CHECK FOR USE OF AUTOREGRESSIVE CONSTANTS
      IF((ITERSW .NE.0).AND.(IAUTOR.EQ.0)) B1 = B1 -AUTOR(I)*BL(I,J)
      IPNO = NBP(I,J)
      IF( IPNO .GT. NOTS ) GO TO 3020
      RES(I) = RES(I) - TSDELT(IPNO) * B1
      GO TO 3032
 3020 CONTINUE
      IPNO = IPNO - NOTS
      IF( IPNO .GT. NOAS ) GO TO 3022
      RES(I) = RES(I) - ASDELT(IPNO) * B1
      GO TO 3032
 3022 CONTINUE
      IPNO = IPNO - NOAS
      RES(I) = RES(I) - PSDELT(IPNO) * B1
 3032 CONTINUE
C     PRINT CALCULATED RESIDUALS
 3034 CONTINUE
 3036 CONTINUE
      READ(NUPASS) NOSTA,NOPASS,L1,L2,SIGMA1,SIGMA2
      WRITE(6,9001) ISTA,STLABL(ISTA),IYR,IMO,IDA,IHR,IMIN,SEC,IPASS,
     * RES(1),RES(2),SIGMA1,SIGMA2,L1,L2
      IF(ITERSW.NE.0) WRITE(NURITE) XMLABL(NMSNO(2)),NMSNO(2),L1,L2,
     * XMLABL(NMSNO(1)),RES(1),SIGMA1,RES(2),SIGMA2
      ICNT= ICNT+1
C     CLEAR ARRAYS
      DO 6008 I=1,NSIZEM
      NMSNO(I) = 0
      NBL(I) = 0
      DO 6008 J= 1,NSIZEB
      BL(I,J) = 0.D0
      NBPL(I,J) =0
 6008 CONTINUE
      GO TO 100
C
C
C     END OF INPUT DATA
C
```

```
 4000 CONTINUE                                                136300
      REWIND ISFILE                                           136380
      REWIND IARCIN                                           136390
      REWIND IPASIN                                           136400
      REWIND IALPHA                                           136410
      REWIND NUPASS
      IF(ITERSW.EQ.0) RETURN
      END FILE NURITE
 4010 READ(NUTAPE,END=4020) ADUM
      WRITE(NURITE) ADUM
      GO TO 4010
 4020 REWIND NURITE
      RETURN
C                                                             136450
C                                                             136460
C     PREMATURE END OF FILE - IARCIN - ERROR                  136470
 4100 CONTINUE                                                136480
      WRITE (6,5801)                                          136490
 4103 CONTINUE                                                136500
      STOP                                                    136510
C                                                             136520
C     PREMATURE END OF FILE - IPASIN - ERROR                  136530
 4101 CONTINUE                                                136540
      WRITE (6,5802)                                          136550
      GO TO 4103                                              136560
      END                                                     136570
                                                          0378 CARDS
```

A-3.6 SPECIAL SUBROUTINE PRTIAL

```
      SUBROUTINE PRTIAL                                                 70980
      IMPLICIT REAL*8 (A-H,O-Z)                                         70990
      LOGICAL*1 ELEVOK,IEND
C                                                                       71000
C VERSION OF 12/05/68 FOR THE NETWORK ANALYSIS PROGRAM (NAP-2)          71010
C       FORTRAN SUBROUTINE                                              71020
C       FOR USE WITH H LEVEL FORTRAN COMPILER ON IBM 360/MOD 95.        71030
C                                                                       71040
C PURPOSE                                                               71050
C       PRTIAL IS THE CONTROL PROGRAM FOR THE CALCULATION OF THE        71060
C       PARTIAL DERIVATIVES, FUNCTIONAL DISCREPANCIES, AND PARAMETER    71070
C       DISCREPANCIES.                                                  71080
C                                                                       71090
C CALLING SEQUENCE                                                      71100
C       CALL PRTIAL                                                     71110
C                                                                       71120
C INPUT                                                                 71130
C                                                                       71140
C                                                                       71150
C OUTPUT                                                                71160
C                                                                       71170
C                                                                       71180
C REFERENCE                                                             71190
C                                                                       71200
C                                                                       71210
C METHOD                                                                71220
C                                                                       71230
C                                                                       71240
C RESTRICTIONS                                                          71250
C                                                                       71260
C                                                                       71270
C ACCURACY                                                              71280
C       FLOATING POINT - ALL CONSTANTS AND VARIABLES ARE CARRIED IN     71290
C                        DOUBLE PRECISION TO 16 DECIMAL DIGITS.         71300
C                        MIN. MAGNITUDE = 1.D-75                        71310
C                        MAX. MAGNITUDE = 1.D+75                        71320
C       FIXED POINT - MAGNITUDE OF ALL CONSTANTS AND VARIABLES ARE      71330
C                     .LE. 2147483647                                   71340
C                                                                       71350
C REQUIRED SUBPROGRAMS - FORTRAN LIBRARY                                71360
C                                                                       71370
C                                                                       71380
C REQUIRED SUBPROGRAMS - OTHER                                          71390
C       ERRORP - CONTROL THE PRINT-OUT OF ERROR MESSAGES                71400
```

```
C         MEASUR - PARTIAL DERIVATIVES WRT MASTER COORDINATE SYSTEM AND -
C                  WRT THE ERROR MODEL COEFFICIENTS
C         REFRCT - REFRACTION CORRECTION
C         ROTFIX - ROTATES VECTOR FROM INERTIAL TO EARTH FIXED COORD.
C         ROTINT - ROTATION MATRIX FROM EARTH-FIXED TO LOCAL STABLE -
C                  COORDINATES.
C   STORAGE REQUIREMENTS
C
C
C   TIMING
C         NO ESTIMATE AVAILABLE
C   ANALYSIS
C         JOE LYNN           ,PROJECT LEADER          ,DBA SYSTEMS INC.
C         RAUL GARZA-ROBLES ,SEN. PROGRAMMER/ANALYST ,      ''
C         MRS. SHIRAS GUION , ''        ''       ''   ,      ''
C         ROBERT DEVANEY    , ''        ''       ''   ,      ''
C
C   PROGRAMMER
C         ROBERT DEVANEY    , ''        ''       ''   ,      ''
C
C
C   PROGRAM MODIFICATIONS
C
C
C
C********** START PROGRAM **********
C
COMMON - ARC INFORMATION
      COMMON /COMSOL/ FREP(500) , IFREP(500),NTSPN,NEGTSP,IRITKN,KOUT
      COMMON /ACINFO/
     1ACLABL(100)    ,EPOCH(100)     ,EPOCHD(100)    ,GHA(100)       ,
     2IROTAT(100)    ,IASTAB(100,6)  ,IAPNO(100,6)
C
COMMON - ARC STABLE PARAMETERS
      COMMON /ASPARM/
     1NOAS           ,NOAS2          ,ASLABL(200)    ,ASINT(200)     ,
     2ASCUR(200)     ,ASIWT(200)     ,ASCWT(200)     ,ASDELT(200)    ,
     3IASEVL(200)    ,AS2LAB(200)    ,AS2INT(200)
C
COMMON - DEBUG PRINTING
C
      COMMON /CDEBUG/
     1IDEFLG(2)      ,IPRFLG(2)      ,ISLFLG(2)
C
```

```
COMMON - PARAMETERS COMPUTED ON INITIAL CALL TO MEASURE
      COMMON /CMEASR/
     1RANGE          ,EL              ,RDOT            ,EDOT           ,
     2SINE           ,COSE
C
COMMON - CONVERSION CONSTANTS LOADED BY MAIN PROGRAM
      COMMON /CONMET/
     1XMETKM         ,XMETFT          ,XMETUD
C
COMMON - WORK AREA
      COMMON /CWORK/
     1IDATA(500)     ,DATA(500)       ,ICTL            ,NINT           ,
     2NFLT
C
COMMON - INTEG I/O
C
      COMMON /CINTEG/ PV(6), VEM(6,6), SV(6), SVEM(6,6), TE, TT
     1     , T2
C
COMMON - GENERAL CONTROL INFORMATION
      COMMON /GENCOM/
     1CONVG          ,DIVRG           ,SUMO            ,SUM1           ,
     2COST           ,MXITER          ,MXDIVG          ,ITERNO         ,
     AITERSW         ,
     3IDIVNO         ,IUNITS          ,IGEOFL          ,IAUTOR         ,
     4IOINP          ,ISIGNO          ,IUPDNO          ,IPRRES         ,
     5NARCS          ,NSTA            ,MXMEAS          ,MXARCS         ,
     6MXPARM         ,MXSTA           ,MXLIN           ,IPGNO          ,
     7                PGID(10)
C
COMMON - INTEG I/O
      COMMON /EXTCM/ XP(10), YP(10), ZP(10), XV(10), YV(10), ZV(10),
     1    BD(10), BT(10), BR(10), BM(10), BG(10), TZ, TF, TJ, TU, DP,
     3    EP, SP, CD, D, V, T, NB, NS, NE, NM, NP, NC, KT, KD, KV, KE,
     3    KS, LA, LB, LW
C
COMMON - I/O FILE ASSIGNMENTS
      COMMON /IONUMB/
     1MFILE          ,ISFILE          ,IARCIN          ,IPASIN         ,
     2IARCOT         ,IPASOT          ,ICARD           ,IRESID         ,
     3ITAPE          ,IALPHA          ,LUA             ,LUB            ,
     4 LUC
C
COMMON - MEASUREMENT INFORMATION
```

```
      COMMON /MSINFO/
     1XMLABL(300)     ,MSTYPE(300)
C
COMMON - PASS STABLE PARAMETERS
      COMMON /PSPARM/
     1NOPS            ,NOPS2            ,PSLABL(200)     ,PSINT(200)     ,
     2PSCUR(200)      ,PSIWT(200)       ,PSCWT(200)      ,PSDELT(200)    ,
     3IPSEVL(200)     ,PS2LAB(200)      ,PS2INT(200)
C
COMMON - STATION INFORMATION
      COMMON /STINFO/
     1STLABL(100)     ,STLAT(100)       ,STLON(100)      ,STHT(100)      ,
     2ISTSUR(100)     ,ISTSID(100)
C
COMMON - TOTALLY STABLE PARAMETERS
      COMMON /TSPARM/
     1NOTS            ,NOTS2            ,TSLABL(200)     ,TSINT(200)     ,
     2TSCUR(200)      ,TSIWT(200)       ,TSCWT(200)      ,TSDELT(200)    ,
     3ITSEVL(200)     ,TS2LAB(200)      ,TS2INT(200)
C
C
      COMMON /EARTH/  ROTAT,  GRHA,  RADIUS,  GRAV,  ECC
C
      COMMON /RSUMR/DRXP(6),DRRXP(6),DRXS(6),DRRXS(6),RSVAL,RSR
C
      DIMENSION IDATAW(100),DATAW(100)
      DIMENSION LO(2),QSUM(2),QSQ(2),TIMSEC(156),QMEAS(156,2)
C
      DIMENSION
     1NOMEAS(30)      ,NRFRC(30)        ,NRC(30)         ,NPMEAS(300)    ,
     2NEMT(300)       ,NOPARM(300)      ,NPTYPE(300)     ,ELMIN(30)      ,
     3ELMAX(30)       ,RINDEX(30)       ,WGTM(30)        ,TLC(3,5)
C
      DIMENSION NMS(30),      DMS(30)
C
      DIMENSION  BDO(6),      BDOM(6),      BDOS(6),      NBDO(6),
     1      NBDOS(6)
C
      DIMENSION XX(6)
      DIMENSION PV1(6), AVVEM(6,6)
C
C  ALL FORMAT STATEMENTS ARE CONTAINED HERE.
    1 FORMAT('1************************'/
     1        ' ***                    ***'/
```

```
     2          ' *** PRTIAL SPEAKING ***'/
     3          ' ***                 ***'/
     4          ' ************************')
    2 FORMAT('0*** STANDARD FILE FORMAT *** - FOR INTERPRETATION REFER T
     1O DOCUMENTATION WITH INDICATED FILE AND ICTL'/
     2'OICTL ='I4/
     3' NINT ='I4/
     4' NFLT ='I4/
     5'OIDATA(I) , I=1,NINT'/(10X,20I5))
    3 FORMAT('ODATA(I)  , I=1,NFLT'/(9X,1P6D18.10))
    8 FORMAT('0    IASTAB('I3,1H,I1,3H) =I2,'WILL CONTINUE PROCESSING WIT
     1H STABILITY FLAG (IASTAB) = 1.')
   10 FORMAT('0    MASTER TABLE OF MEAS. NUMBERS DOES NOT CONTAIN MEAS. N
     1O. FOR THIS POINT.  PRTIAL WILL IGNORE THIS PT.'/
     2'     NMS(N) ='I3)
   11 FORMAT('0*** ISFILE OUTPUT RECORD ***')
   12 FORMAT('0     NMEAS AND/OR NPARMS ARE TOO LARGE.  PRTIAL CANNOT PRO
     1CEED.'/5X,'NMEAS ='I5,'  , NPARMS ='I5)
   13 FORMAT(' E = ',D20.12,' E PRIME = ',D20.12/
     1         ' F = ',D20.12,' F PRIME = ',D20.12/
     2         ' G = ',D20.12,' G PRIME = 'D20.12)
   15 FORMAT(' TIME',D20.10/' XYZ ',6D20.10/(6D20.10))
   17 FORMAT(' COS(DEC) AND COS(EL) ',2D20.10)
C
C      SET UP C = SPEED OF LIGHT IN APPROPIATE UNITS
C
       C = 2.997925D8
       IF( IUNITS .EQ. 1 ) C = C * XMETKM
       IF( IUNITS .EQ. 2 ) C = C * XMETFT
       IF( IUNITS .EQ. 3 ) C = C * XMETUD
C
C
       ZERO= 0.D0
       NUTAPE= 35
       NUPASS= 36
       NURITE=37
       IF(ITERNO.LE.1) REWIND ITAPE
       REWIND NURITE
C
C   ALL REFERENCES TO IPRFLG(I) .NE. 0 EXERCISES DEBUG PRINTOUT
       IF((IPRFLG(1).NE.0).AND.(IPRFLG(2).GE.ITERNO)) WRITE (6,1)
C
C   INPUT MFILE - ARC NUMBER
       READ (MFILE)
```

```
     1      ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),(DATA(I),I=1,NFLT)
 1000 CONTINUE
      NOARC = IDATA(1)
C
C  STORE INFORMATION FOR INTEGRATOR IN /EXTCM/
C
      NOSEC = IDATA(2)
      NB = IDATA(3)
      NS = IDATA(4)
      NE = IDATA(5)
      NM = IDATA(6)
      NP = IDATA(7)
      NC = IDATA(8)
      KT = IDATA(9)
      KD = IDATA(10)
      KV = IDATA(11)
      KE = IDATA(12)
      KS = IDATA(13)
      LA = IDATA(14)
      LB = IDATA(15)
      LW = IDATA(16)
      TZSEC = DATA(2)
      TZDAY = DATA(1)
      TFSEC = DATA(4)
      TFDAY = DATA(3)
      DO 1003 I=1,10
      XP(I) = DATA(I+4)
      YP(I) = DATA(I+14)
      ZP(I) = DATA(I+24)
      XV(I) = DATA(I+34)
      YV(I) = DATA(I+44)
      ZV(I) = DATA(I+54)
      BD(I) = DATA(I+64)
      BT(I) = DATA(I+74)
      BR(I) = DATA(I+84)
      BM(I) = DATA(I+94)
      BG(I) = DATA(I+104)
 1003 CONTINUE
      TZ = DATA(115)
      TF = DATA(116)
      TJ = DATA(117)
      TU = DATA(118)
      DP = DATA(119)
      EP = DATA(120)
```

```
      SP = DATA(121)
      CD = DATA(122)
      D = DATA(123)
      V = DATA(124)
      T = DATA(125)
      XPOLAR = DATA(126)
      YPOLAR = DATA(127)
C
C
C
C  OUTPUT ISFILE - ARC RECORD - ICTL=1
      NINT = 1
      NFLT = 1
      WRITE (ISFILE)
     1     ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),(DATA(I),I=1,NFLT)
C
      READ (MFILE) ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),(FREP(J),J=1,NFLT)
      NTSPN = IDATA(1)
      NEGTSP = IDATA(2)
      DO 1005 I = 1,500
 1005 IFREP(I) = 0
      IRITKN = 0
      KOUT = 0
C  INPUT IARCIN - ARC STABLE PARAMETERS
      READ (IARCIN)
     1     ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),(DATA(I),I=1,NFLT)
 1010 IF(ICTL .EQ. 1) GO TO 1020
C
C  STORE ARC STABLE PARAMETERS NOT ENTERING INTO SOLUTION
      NOAS2 = IDATA(2)
      DO 1011 I = 1,NOAS2
      K = 2*I - 1
      AS2LAB(I) = DATA(K)
 1011 AS2INT(I) = DATA(K+1)
C
C  INPUT IARCIN - ARC STABLE PARAMETERS
      READ (IARCIN)
     1     ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),(DATA(I),I=1,NFLT)
 1020 NOAS = IDATA(2)
      IF(NOAS .EQ. 0) GO TO 1050
C
C  STORE ARC STABLE PARAMETERS ENTERING INTO SOLUTION
      DO 1030 I = 1,NOAS
      K = 6*I - 5
```

```
      ASLABL(I) = DATA(K)
      ASINT(I) = DATA(K+1)
      ASCUR(I) = DATA(K+2)
      ASIWT(I) = DATA(K+3)
      ASCWT(I) = DATA(K+4)
      ASDELT(I) = DATA(K+5)
 1030 IASEVL(I) = IDATA(I+2)
C
C
C
C  OBTAIN STATE VECTORS AND GENERATE POWER SERIES
C
C
 1050 CONTINUE
      IF( NOSEC .LE. 0 ) GO TO 1060
C
C  SECONDARY ARC
C
      REWIND LUC
      N = NOSEC
      LW = 1
      LB = LUC
      IRET = 1
      GO TO 1900
 1060 CONTINUE
      IRTKN1 = IRITKN
C
C  PRIMARY ARC
C
      REWIND LUB
      N = NOARC
      LW = 1
      LB = LUB
      IRET = 2
      GO TO 1900
 1070 CONTINUE
      KOUT = IRTKN1
      IPS1 = 1
      IPS2 = 1
C
C  INPUT MFILE - PASS/STATION RECORD - ICTL=2
      READ (MFILE)
     1     ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),(DATA(I),I=1,NFLT)
2000   CONTINUE
```

```
C
C   STORE PASS INFORMATION
 2010 NOSTA = IDATA(2)
      NOPASS = IDATA(3)
      NMEAS = IDATA(4)
      NPARMS = IDATA(5)
      IF((NMEAS .LE. 30) .AND. (NPARMS .LE. 300)) GO TO 2018
C
C  *** ERROR CONDITION
2014   CONTINUE
      WRITE(6,12) NMEAS,NPARMS
      GO TO 7100
 2018 DO 2020 I = 1,NMEAS
      LO(I)=0
      QSUM(I)=0.D0
      QSQ(I)=0.D0
      K = 3*I + 3
      NOMEAS(I) = IDATA(K)
      NRFRC(I) = IDATA(K+1)
      NRC(I) = IDATA(K+2)
      K = 4*I - 3
      ELMIN(I) = DATA(K)
      ELMAX(I) = DATA(K+1)
      RINDEX(I) = DATA(K+2)
 2020 WGTM(I) = DATA(K+3)
C
C   STORE PARAMETER TABLE
      ITEMP = 3 * NMEAS
      DO 2025 I = 1,NPARMS
      K = 4*I + ITEMP - 3
      NPMEAS(I) = IDATA(K+5)
      NEMT(I) = IDATA(K+6)
      NOPARM(I) = IDATA(K+7)
 2025 NPTYPE(I) = IDATA(K+8)
      L=0
C
C   INPUT IPASIN - PASS STABLE PARAMETERS
      READ (IPASIN)
     1     ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),(DATA(I),I=1,NFLT)
 2100 IF(ICTL .EQ. 1) GO TO 2200
C
C   STORE PASS STABLE PARAMETERS NOT ENTERING INTO SOLUTION
      NOPS2 = IDATA(4)
      DO 2110 I = 1,NOPS2
```

```
      K = 2*I - 1
      PS2LAB(I) = DATA(K)
 2110 PS2INT(I) = DATA(K+1)
C
C  INPUT IPASIN - PASS STABLE PARAMETERS
      READ (IPASIN)
     1      ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),(DATA(I),I=1,NFLT)
 2200 NOPS = IDATA(4)
      IF(NOPS .EQ. 0) GO TO 2250
C
C  STORE PASS STABLE PARAMETERS ENTERING INTO SOLUTION
      DO 2230 I = 1,NOPS
      K = 6*I - 5
      PSLABL(I) = DATA(K)
      PSINT(I) = DATA(K+1)
      PSCUR(I) = DATA(K+2)
      PSIWT(I) = DATA(K+3)
      PSCWT(I) = DATA(K+4)
      PSDELT(I) = DATA(K+5)
 2230 IPSEVL(I) = IDATA(I+4)
C
C  OUTPUT ISFILE (FOR SOLVER) - PASS RECORD - ICTL=2
 2250 ICTL = 2
      NINT = NMEAS + 4
      NFLT = NMEAS
      IDATA(4) = NMEAS
      DO 2275 I = 1,NMEAS
      IDATA(I+4) = NOMEAS(I)
 2275 DATA(I) = WGTM(I)
      WRITE (ISFILE)
     1      ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),(DATA(I),I=1,NFLT)
C
C  ROTATION MATRIX FROM EARTH-FIXED TO LOCAL STABLE COORDINATES
C                = TLC(1,1) TO TLC(3,3)
C
C  TRANSLATION VECTOR
C                = TLC(1,4) TO TLC(3,4)
      CALL ROTINT(STLAT(NOSTA),STLON(NOSTA),STHT(NOSTA),TLC)
C
C  APPLY POLAR MOTION IF DEFAULTS OF 0.0D0 ON X,Y ANGLES WERE OVERRIDDEN
C
      E = TLC(1,4)
      F = TLC(2,4)
      G = TLC(3,4)
```

```
      TLC(1,4) = E - XPOLAR * G
      TLC(2,4) = F + YPOLAR * G
      TLC(3,4) = G + XPOLAR * E - YPOLAR * F
      IF ((IPRFLG(1) .EQ. 0) .OR. (ITERNO .GT. 1))GO TO 2280
      WRITE (6,13) E,TLC(1,4) , F,TLC(2,4) , G,TLC(3,4)
 2280 CONTINUE
      SAVE1 = TLC(1,1)
      SAVE2 = TLC(2,1)
C
C
C  STATION OFFSETS
C                   = TLC(1,5) TO TLC(3,5)
      DO 2300 I = 1,3
 2300 TLC(I,5) = 0.D0
C
C  SEARCH ERROR MODEL TABLE FOR SURVEY CORRECTIONS , IF ANY
C
      DO 2500 I=1,NPARMS
      IF((NEMT(I).LT.7).OR.(NEMT(I).GT.9)) GO TO 2500
      K= NPTYPE(I)
      GO TO (2410,2420,2430,2440,2450,2460), K
 2410 TLC(NEMT(I)-6,5) = TSCUR(NOPARM(I))
      GO TO 2500
 2420 TLC(NEMT(I)-6,5) = ASCUR(NOPARM(I))
      GO TO 2500
 2430 TLC(NEMT(I)-6,5) = PSCUR(NOPARM(I))
      GO TO 2500
 2440 TLC(NEMT(I)-6,5) = TS2INT(NOPARM(I))
      GO TO 2500
 2450 TLC(NEMT(I)-6,5) = AS2INT(NOPARM(I))
      GO TO 2500
 2460 TLC(NEMT(I)-6,5) = PS2INT(NOPARM(I))
 2500 CONTINUE
C
C  INPUT MFILE - MEASUREMENT POINT RECORD - ICTL=3
 7000 CONTINUE
      READ (MFILE,END=7100)
     1     ICTL,NINT,NFLT,(IDATA(I),I=1,NINT),(DATA(I),I=1,NFLT)
      GO TO (5100,5100,3000),ICTL
C
C  *** NORMAL EXIT ***
C
 7100 IEND=.TRUE.
      IF(L.GT.0) GO TO 5110
```

```
 7105 REWIND ISFILE
      REWIND NUTAPE
      REWIND NUPASS
      REWIND MFILE
      REWIND IARCIN
      REWIND IPASIN
      RETURN
C
C  PROCESS MEASUREMENT POINT DATA
 3000 CONTINUE
C
C  STORE MEASUREMENT POINT DATA
 3010 NMSPT = NINT
      DO 3020 I = 1,NMSPT
      NMS(I) = IDATA(I)
 3020 DMS(I) = DATA(I+2)
C
C  DTARR = DELTA TIME FOR AVERAGE RANGE RATE DATA
C
      DTARR = DATA (NMSPT + 3)
C
C
C  CURRENT TIME OF MEAS. PT. (JULIAN DAYS SINCE 1950)
      TDCURR = DATA(1)
C
C  TIME IN SECONDS OF GIVEN DAY
      TSCURR = DATA(2)
C
      L=L+1
      TIMSEC(L)=TSCURR
      DO 3030 I=1,NMEAS
 3030 QMEAS(L,I) = 1.D50
 3040 CONTINUE
C  T FOR NAP  INTEG  = TIME(SEC) FROM MIDNIGHT OF EPOCH DAY
      T1 = EPOCH(NOARC)
      DTDAY = TDCURR - EPOCHD(NOARC)
      T2 = DTDAY * 86400.D0 + TSCURR
C
C
C
C  SOLVE FOR VECTOR AND MATRIZANT
C
C
      TT= TDCURR*86400.D0 + TSCURR
```

```
      TMP = TT - TJ
C  PRIMARY ARC
C
 3050 CONTINUE
      CALL SOFORT(IPS1,LUB,0.D0,TMP,D,V,T,PV,VEM)
C
C
      IPS1 = 2
      GRHA = GHA ( NOARC)
C
C  ROTATE STATE VECTOR FROM INERTIAL TO EARTH-CENTERED/EARTH FIXED COORD
      CALL ROTFIX(PV,T2)
C
      NPAK = 0
C
C  SAVE PARAMETERS THAT WOULD BE DESTROYED BY MEASUR
 3270 CONTINUE
      SAVE3 = PV(1)
C  ***
C  ***  MAIN LOOP - PROCESS EACH MEASUREMENT FOR THIS POINT
C  ***
      IF(NMSPT.LE.0) GO TO 6100
      DO 4000 N = 1,NMSPT
      DO 3200 I = 1,NMEAS
      IF(NMS(N) .NE. NOMEAS(I)) GO TO 3200
      MINDEX = I
      GO TO 3250
 3200 CONTINUE
C  *** ERROR CONDITION
C
C  MEAS. NO. FOR THIS PT DOES NOT MATCH MASTER TABLE OF MEAS. NUMBERS
```

```
      WRITE(6,10) NMS(N)
      GO TO 4000
C
C  RESTORE PARAMETERS THAT ARE DESTROYED BY MEASUR
C
 3250 CONTINUE
      TLC(1,1) = SAVE1
      TLC(2,1) = SAVE2
      PV(1) = SAVE3
      MS = MSTYPE(NOMEAS(MINDEX))
C
C  AVERAGE RANGE-RATE DATA (MYTPE=18)
C  PRIMARY ARC ONLY
C
      IF (MS .NE. 18)  GO TO 3254
      RPLUS = 0.0D0
      RMINUS = 0.0D0
      DTARR2 = DTARR/2.D0
      TMP1 = TMP + DTARR2
C
      CALL SOFORT(IPS1,LUB,0.0D0,TMP1,D,V,T,PV1,AVVEM)
      T3 = T2 + DTARR2
      CALL ROTFIX(PV1,T3)
C  COMPUTE RANGE FOR TIME + DT/2  (RPLUS)
      DO 3252 I=1,3
      TMP1 = PV1(I) - TLC(I,4)
      RPLUS = RPLUS + TMP1 * TMP1
 3252 CONTINUE
      RPLUS = DSQRT (RPLUS)
C
      TMP1 = TMP - DTARR2
C
      CALL SOFORT(IPS1,LUB,0.0D0,TMP1,D,V,T,PV1,AVVEM)
      T3 = T2/- DTARR2
      CALL ROTFIX(PV1,T3)
C
C  COMPUTE RANGE FOR TIME - DT/2  (RMINUS)
```

```
      VALUE = (RPLUS - RMINUS ) / DTARR
C
3254   CONTINUE
C
C
C
C  MEASUR RETURNS RANGE,EL,RDOT,EDOT,SINE,COSE FOR JTERM = 0
C
      CALL MEASUR( MS, 0, TLC, PV, VALUE )
C
C  SAVE COS(DEC)
C
      COSDEC = VALUE
      VALUE = 0.0D0
C
C  COMPUTE COS(ELEV)
      COSEL = DCOS(EL)
       IF ((IPRFLG(1).GE.2).AND.(ITERNO.LE.IPRFLG(2))) WRITE(6,17)
     1 COSDEC,COSEL
C
C  CHECK FOR ACCEPTABLE ELEVATION
C
C
C
      ELEVOK=.TRUE.
      IF((EL.LT.ELMIN(MINDEX)).OR.(EL.GT.ELMAX(MINDEX))) ELEVOK=.FALSE.
C
C  ELEVATION OK
 3300 CONTINUE
C
C  CLEAR ARRAYS FOR SAVING PARTIAL DERIVATIV ES OF VECTOR
      DO 3290 I=1,6
      NBDO(I) = 0
      NBDOS(I) = 0
      BDO(I) = 0.D0
      BDOS(I) = 0.D0
 3290 CONTINUE
C
C  CALL MEASUR WITH JTERM=1 TO GET CALCULATED MEASUREMENT
      CALL MEASUR( MS, 1, TLC, PV, VALUE )
C
      DO 3500 I = 1,NPARMS
      IF(NOMEAS(MINDEX) .NE. NPMEAS(I)) GO TO 3500
C
```

```
C  FOR ERROR MODEL TERMS GE 10, FIND CURRENT PARAMETER VALUE
C
      IF( NEMT(I) .EQ. 1 ) GO TO 3390
      IF (NEMT(I) .LT. 10) GO TO 3380
      K = NPTYPE(I)
      GO TO (3310,3320,3330,3340,3350,3360), K
 3310 CONTINUE
      TLC(1,1) = TSCUR(NOPARM(I))
      GO TO 3380
 3320 CONTINUE
      TLC(1,1) = ASCUR(NOPARM(I))
      GO TO 3380
 3330 CONTINUE
      TLC(1,1) = PSCUR(NOPARM(I))
      GO TO 3380
 3340 CONTINUE
      TLC(1,1) = TS2INT(NOPARM(I))
      GO TO 3380
 3350 CONTINUE
      TLC(1,1) = AS2INT(NOPARM(I))
      GO TO 3380
 3360 CONTINUE
      TLC(1,1) = PS2INT(NOPARM(I))
 3380 CONTINUE
      CALL MEASUR( MS, NEMT(I), TLC, PV, VALUE )
C
C  CHECK FOR PARAMETER EVALUATION TYPE 2 - NOT TRANSMITTED TO SOLVER
C         NPTYPE(I) = 4  TOTALLY STABLE - NOT ENTERING SOLUTION
C                   = 5  ARC STABLE - NOT ENTERING SOLUTION
C                   = 6  PASS STABLE - NOT ENTERING SOLUTION
C
 3390 CONTINUE
      IF(L.GT.0) GO TO 3500
      IF(.NOT.ELEVOK) GO TO 3500
      IF(NPTYPE(I) .GT. 3) GO TO 3500
      IF (MS .EQ. 2) PV(1) = PV(1) * COSEL
       IF (MS .EQ. 4) PV(1) = PV(1) * COSDEC
      NPAR = NPAR + 1
      ITEMP = NPAR + NMSPT
      ITEMP1 = 3*NPAR + NMSPT - 2
C
C  IF PARTIAL OF VECTOR WRT MEAS, SAVE IT FOR ROTATION IT INERTIAL COORD
C
      IF(NEMT(I) .GT. 6 ) GO TO 3400
```

```
      NBDO(NEMT(I)) = ITEMP + 2
      BDO(NEMT(I)) = PV(1)
 3400 CONTINUE
C
C
C  PARAMETER
      IDATAW(ITEMP1+2) = NPTYPE(I)
C  PARAMETER NUMBER
      IDATAW(ITEMP1+3) = NOPARM(I)
C  MEASUREMENT NUMBER
      IDATAW(ITEMP1+4) = NPMEAS(I)
C  PARTIAL
      DATAW(ITEMP + 2) = PV(1)
 3500 CONTINUE
C
C  IF NO PARTIALS OF VECTOR, SKIP ROTATION
      IF( NBDO(1) .LE.0 ) GO TO 3540
C
C  ROTATE PARTIALS OF VECTOR FROM FIXED TO INERTIAL COORDINATES
      CALL ROTPAR( BDO, T2 )
C
C  RELATE VECTOR TO EPOCH TIME
C
      DO 3520 K=1,6
      BDOM(K) = 0.D0
      DO 3520 J=1,6
 3520 BDOM(K) = BDOM(K) + BDO(J) * VEM(J,K)
C
      DO 3530 J =1,6
 3530 DATAW(NBDO(J)) = BDOM(J)
C
 3540 CONTINUE
C
C  CHECK FOR PARTIALS OF SECONDARY VECTOR
      IF( NBDOS(1) .LE. 0 ) GO TO 3570
C  ROTATE TO INERTIAL
      CALL ROTPAR( BDOS, T2 )
C  RELATE TO EPOCH VIA MATRIZANT
      IF ((IPRFLG(1).GE.2) .AND. (ITERNO .LE. IPRFLG(2))) WRITE (6,15)
     1   T,(SV(I),I=1,6),((SVEM(I,J),I=1,6),J=1,6)
      DO 3550 K=1,6
      BDOM(K) = 0.D0
      DO 3550 J=1,6
 3550 BDOM(K) = BDOM(K) + BDOS(J) * SVEM(J,K)
```

```
C
      DO 3560 J=1,6
 3560 DATA(NBDOS(J)) = BDOM(J)
C
 3570 CONTINUE
C
C
C  CALL MEASUR WITH ACTUAL MEASUREMENT VALUE FOR CORRECT SIGN FOR
C          ANGULAR DATA
C
C
      PV(1) = DMS(N)
      CALL MEASUR( MS, -1, TLC, PV, VALUE )
      IF(NRFRC(MINDEX) .EQ. 0) GO TO 3600
C
C  REFRACTION CORRECTION
      CALL REFRCT(RINDEX(MINDEX),STLAT(NOSTA),MS,CORR)
      VALUE = VALUE + CORR
C
C  MEASUREMENT NUMBERS
 3600 IF (L.LE.0) GO TO 6000
C
C  MEASUREMENT DISCREPANCIES
      DATA(N + 2) = DMS(N) - VALUE
      IF (IOINP .EQ. 6) DATA(N+2) = -DMS(N)
      IF ((IOINP .LT. 7).AND.(MS.EQ.2)) DATA(N+2) = DATA(N+2) * COSEL
      IF ((IOINP .LT. 7).AND.(MS.EQ.4)) DATA(N+2) = DATA(N+2) * COSDEC
      IF(L.LE.0) GO TO 6000
      QMEAS(L,MINDEX) = DATA(N+2)
      QSUM(MINDEX) = QSUM(MINDEX) + DATA(N+2)
      QSQ(MINDEX)= QSQ(MINDEX) + DATA(N+2)*DATA(N+2)
      LO(MINDEX)= LO(MINDEX) + 1
 4000 CONTINUE
      IF(L.GT.0) GO TO 7000
C  OUTPUT ISFILE - MEASUREMENT RECORD - ICTL=3
 4010 CONTINUE
      JCTL=3
      JFLT=NMSPT+2+NPAR
      JINT=JFLT+2*NPAR
      IDATAW(1)=NMSPT
      IDATAW(2)=NPAR
      WRITE(ISFILE) JCTL,JINT,JFLT,(IDATAW(I),I=1,JINT),(DATAW(I),I=1,
     *JFLT)
      DATAW(1)= NOARC
```

```
      DATAW(2)= ISTSID(NOSTA)
      DATAW(3)= NOPASS
      IF(LO(1).GT.0) GO TO 4020
      DMS(2)= DMS(1)
      DMS(1)= 0.D0
      QSQ(1)= 0.D0
 4020 IF(LO(2).GT.0) GO TO 4030
      DMS(2)= 0.D0
      QSQ(2)= 0.D0
 4030 WRITE(NUTAPE) ZERO,DATAW(1),DATAW(2),DATAW(3),ZERO,ZERO
      WRITE(NUTAPE) TDCURR,TSCURR,DMS(1),DMS(2),ZERO,ZERO
      WRITE(NUPASS) NOSTA,NOPASS,LO,QSQ
      WRITE(NURITE)NOARC,NOPASS,ISTSID(NOSTA),STLABL(NOSTA),
     *ELEVOK,LMEAN,LA,QSQ,DMS(1),DMS(2),TDCURR,TSCURR,
     *(TIMSEC(L),L=1,LA),((QMEAS(L,I),L=1,LA),I=1,2)
      IF(IEND) GO TO 7105
      GO TO 5105
C                                                                  79060
 5100 IEND=.FALSE.
      IF(L.GT.0) GO TO 5110
 5105 IF(ICTL-1) 1000,1000,2000
 5110 LMEAN=(L+1)/2
      NMSPT=0
      DO 5140 LL=1,NMEAS
      IF(LO(LL).LE.0) GO TO 5140
      NMSPT= NMSPT+1
      NMS(NMSPT)= NOMEAS(LL)
      QSUMT= QSUM(LL)
      QSQT= QSQ(LL)
 5120 IF(LO(LL).LE.1) GO TO 5150
      QL= LO(LL)
      LA=0
      QBAR= QSUMT/QL
      SIGMAT= DSQRT((QSQT-QSUMT*QBAR)/(QL-1.D0))
      TWOSIG=3.D0*SIGMAT
      QSQT= QSQ(LL)
      QSUMT= QSUM(LL)
      DO 5130 LB=1,L
      IF(DABS(QMEAS(LB,LL)-QBAR).LE.TWOSIG) GO TO 5125
      IF(QMEAS(LB,LL).GE.0.1D50) GO TO 5130
      QSUMT= QSUMT- QMEAS(LB,LL)
      QSQT= QSQT- QMEAS(LB,LL)*QMEAS(LB,LL)
      GO TO 5130
 5125 LA=LA+1
```

```
 5130 CONTINUE
      IF(LA.GE.LO(LL)) GO TO 5135
      LO(LL)= LA
      GO TO 5120
 5135 QSUM(LL)=QBAR
      QSQ(LL)= SIGMAT
 5140 CONTINUE
      IF(TSCURR.LT.TIMSEC(LMEAN)) TDCURR=TDCURR-1.DO
      TSCURR=TIMSEC(LMEAN)
      DATAW(1)=TDCURR
      DATAW(2)=TSCURR
      LA=L
      L=0
      GO TO 3040
 5150 QSQ(LL)=0.DO
      GO TO 5140
 6000 DMS(N)=QSUM(MINDEX)
      IF(MS.EQ.2) DMS(N)= DMS(N)/COSEL
      IF(MS.EQ.4) DMS(N)=DMS(N)/COSDEC
      DMS(N)=DMS(N) + VALUE
      DATAW(N+2)=QSUM(MINDEX)
      IDATAW(N+2)=NMS(N)
      IF(ELEVOK) GO TO 4000
      DATAW(N+2)=0.DO
      IDATAW(N+2)=0
      GO TO 4000
C
 6100 NMSPT=1
      IDATAW(3)=0
      DATAW(3)=0.DO
      GO TO 4010
C
C
C  ROUTINE TO PICK UP STATE VECTORS AND GENERATE POWER SERIES
C        N = ARC NO.
C
C
 1900 CONTINUE
      BG(NC) = GHA(N) + EPOCH(N) * ROTAT
      REWIND LUA
      DO 1960 I=1,6
      XX(I) = 0.DO
      ITEMP = IASTAB(N,I)
      GO TO (1920,1930,1910,1940,1950,1910),ITEMP
```

```
C
C  ERROR
1910   CONTINUE
      WRITE (6,8) N,I,ITEMP
      GO TO 1960
C
C  TOTALLY STABLE ENTERING INTO SOLN
C
 1920 XX(I) = TSCUR(IAPNO(N,I))
      GO TO 1960
C
C  ARC STABLE ENTERING INTO SOLN
C
 1930 XX(I) = ASCUR(IAPNO(N,I))
      GO TO 1960
C
C  TOTALLY STABLE NOT IN SOLN
C
 1940 XX(I) = TS2INT(IAPNO(N,I))
      GO TO 1960
C
C  ARC STABLE NOT IN SOLN
C
 1950 XX(I) = AS2INT(IAPNO(N,I))
 1960 CONTINUE
C
C  STORE IN /EXTCM/
C
      XP(NP) = XX(1)
      YP(NP) = XX(2)
      ZP(NP) = XX(3)
      XV(NP) = XX(4)
      YV(NP) = XX(5)
      ZV(NP) = XX(6)
C
C  TJ IS EPOCH TIME (1950.0) IN SECONDS
C
      TJ = EPOCHD(N)*86400.D0 + EPOCH(N)
      TE = TJ
C
C  TZ IS INTEG START TIME
C  TF IS INTEC STOP TIME
C
      TZ = 0.D0
```

```
      TSTART = TZSEC + TZDAY * 86400.D0
      TFINAL = TFSEC + TFDAY * 86400.D0
C
C
      IF( TSTART .GE. TJ ) GO TO 1970
      IF( TJ .GE. TFINAL) GO TO 1980
C
C  ARC SPANS EPOCH -- MUST INTEG TWO SUBARCS
C
      TF = TFINAL - TJ
      CALL KICKER(2)
      TF = TSTART - TJ
      CALL KICKER(2)
      GO TO 1990
C
C  ARC FOLLOWS EPOCH
C
 1970 CONTINUE
      TF = TFINAL - TJ
      CALL KICKER(2)
      GO TO 1990
C
C  ARC PRECEDES EPOCH
C
 1980 CONTINUE
      TF = TSTART - TJ
      CALL KICKER(2)
 1990 CONTINUE
      GO TO (1060,1070), IRET
C
C  END OF POWER SERIES ROUTINE
C
      END
```

0385 CARDS

Appendix A-4

Constraints Imposed on
NAP-II Deck Setup

A-4.1 CONSTRAINTS IMPOSED BY MINITRACK PREPROCESSOR

There are constraints imposed on the NAP-II deck setup by the preprocessing program. This appendix is an attempt to outline the constraints in a concise, quick-reference form.

The Minitrack Preprocessor (Section 2.2.3) imposes an order to the stations in NAP-II (Key 1, Category 201), depending on the order the Minitrack data is preprocessed (data set 5 of Minitrack Preprocessor). The station ID's required by the NAP-II program are generated from the fifty-fifth and fifty-sixth characters of the Minitrack message format (Section 2.2.1). This code is used for the equatorial data, and the code plus 100 is used for the polar data.

Example: Station constants for Minitrack processor are in the following order:

Ft. Meyers

Quito

Lima

Santiago

Newfoundland.

The following information would be generated by the preprocessor for each station:

	Equatorial		Polar	
	Sta. ID	Sta. No.	Sta. ID	Sta. No.
Ft. Meyers	03	1	103	2
Quito	05	3	105	4
Lima	06	5	106	6
Santiago	08	7	108	8
Newfoundland	12	9	112	10

However, if the station's constants have been in order of:

Newfoundland

Santiago

Lima

Quito

Ft. Meyers,

then the information generated would be

	Equatorial		Polar	
	Sta. ID	Sta. No.	Sta. ID	Sta. No.
Newfoundland	12	1	112	2
Santiago	08	3	108	4
Lima	06	5	106	6
Quito	05	7	105	8
Ft. Meyers	03	9	103	10

The station numbers are required on the NAP-II cards (Key 1 of Category 301 and Key 2 of Category 201 and 202) punched by the preprocessor program. The station identification is required in Keys 3 and 4 of the Category 301 card.

Pass numbers are also generated in the preprocessor and are punched in Key 3 of the Category 201, 202, and 999 cards.

In summary, the following information is generated for the Minitrack Preprocessor for the NAP-II program.

A.	STATION ID	Category 301	Key 4
B.	STATION NUMBER	Category 301	Key 1

1) Even number for polar
2) Odd number for equatorial

C.	Start and Stop times of the data pass	Category 201 and 202
D.	End of pass control card	Category 999.

A-4.2 CONSTRAINTS IMPOSED BY THE PRE-NAP PROGRAM

This program is used to modify, add, or delete NAP-II control cards.

The following are the instructions for its usage:

INS I1	Inserts cards after sequence number I1.
DEL I1 I2	Deletes cards between (and including) sequence numbers I1 and I2.
REP I1 I2	Deletes cards between (and including) sequence numbers I1 and I2, and then inserts cards
MOD I1 I2	Modifies cards between (and including) sequence numbers I1 and I2.

Format for above four cards (A4, x, I5, 5x, I5) note that INS, DEL, REP, and MOD start in column 1 and that column 2 should be blank.

INS and REP must be followed by the cards to be inserted. The cards must be followed by a blank card to signify the end of the particular set of inserted cards.

The MOD cards must be followed by a modifying card. Columns not to be modified should be left blank.

When using the computed mid-point Minitrack data, the Category 704 cards have to be deleted from the control cards. The "704" cards should be used only when the entire data spans are used.

A-4.3 CONSTRAINTS IMPOSED ON NAP-II SETUP

A very basic assumption made when processing Minitrack data is that a good estimate of the state vector is available. This is the basis of the whole process.

The a priori sigma information should be realistically assigned with respect to the error model terms and the observed data.

When processing with the "single-data, point pen pass" data file, be sure to delete Category 704 cards that were used for the processing of the entire observed data file.

Measurement numbers (Category 701) have a limitation as to their assignment. Odd measurement numbers are reserved for cosine alpha (ℓ) measurements, and even numbers are reserved for cosine beta (m) measurements. This limitation is imposed by the Post-NAP processor.

Error model term definition cards (Category 601 cards) must be in order of totally stable, arc stable, and pass stable. There can be only one 601 card for each error model term. Duplicates inserted to override a previous 601 card are not permitted.

A-4.4 ADDITIONAL CONSTRAINTS

This section is reserved for constraints not mentioned above that have not come to light. If, in the course of reducing Minitrack data, using the procedure outlined in this report, new constraints are found, please note them in this section.